Challenges of integrating classical numerical simulations and DL-based image processing

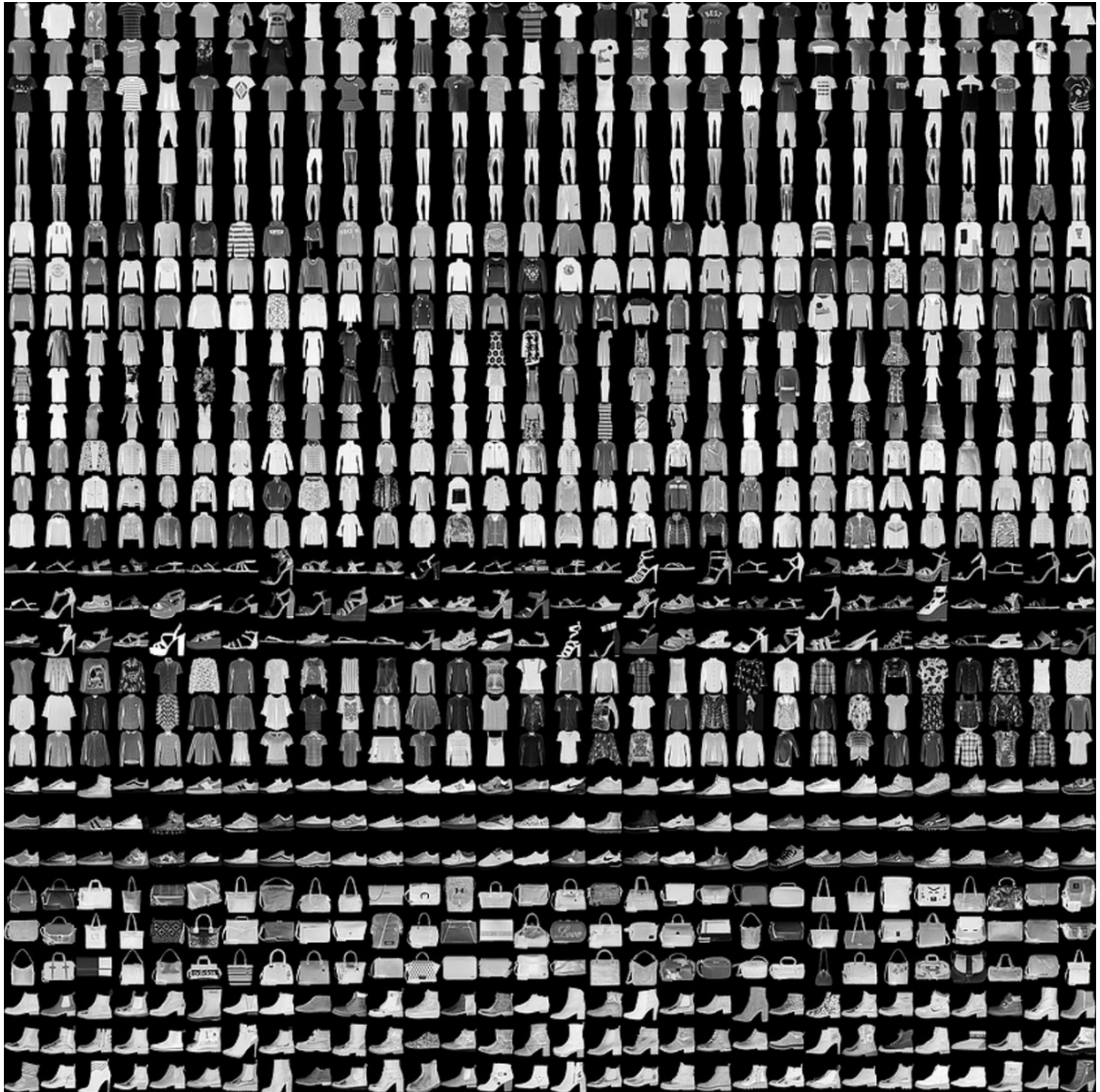
Heuna Kim

What's up for today?

- **A kind archeology tour**
- Examples in Meteorology
- Examples in Eye Tracking

Let’s start with a little bit of Archeology

MNIST



The most valid approach in 2021

Import TensorFlow into your program:

```
import tensorflow as tf

from tensorflow.keras.layers import Dense, Flatten, Conv2D
from tensorflow.keras import Model
```

Load and prepare the MNIST dataset.

```
mnist = tf.keras.datasets.mnist
```

```
from torchvision import datasets
from torchvision.transforms import ToTensor

train_data = datasets.MNIST(
    root = 'data',
    train = True,
    transform = ToTensor(),
    download = True,
)

test_data = datasets.MNIST(
    root = 'data',
    train = False,
    transform = ToTensor()
)
```

yann.lecun.com/exdb/mnist/

THE MNIST DATABASE

of handwritten digits

Yann LeCun, Courant Institute, NYU
Corinna Cortes, Google Labs, New York
Christopher J.C. Burges, Microsoft Research, Redmond

Please refrain from accessing these files from automated scripts with high frequency. Make copies!`

CLASSIFIER	PREPROCESSING	TEST ERROR RATE (%)	Reference
Linear Classifiers			
linear classifier (1-layer NN)	none	12.0	LeCun et al. 1998
linear classifier (1-layer NN)	deskewing	8.4	LeCun et al. 1998
pairwise linear classifier	deskewing	7.6	LeCun et al. 1998
K-Nearest Neighbors			
K-nearest-neighbors, Euclidean (L2)	none	5.0	LeCun et al. 1998
K-nearest-neighbors, Euclidean (L2)	none	3.09	Kenneth Wilder, U. Chicago
K-nearest-neighbors, L3	none	2.83	Kenneth Wilder, U. Chicago
K-nearest-neighbors, Euclidean (L2)	deskewing	2.4	LeCun et al. 1998
K-nearest-neighbors, Euclidean (L2)	deskewing, noise removal, blurring	1.80	Kenneth Wilder, U. Chicago
K-nearest-neighbors, L3	deskewing, noise removal, blurring	1.73	Kenneth Wilder, U. Chicago
K-nearest-neighbors, L3	deskewing, noise removal, blurring, 1 pixel shift	1.33	Kenneth Wilder, U. Chicago
K-nearest-neighbors, L3	deskewing, noise removal, blurring, 2 pixel shift	1.22	Kenneth Wilder, U. Chicago
K-NN with non-linear deformation (IDM)	shiftable edges	0.54	Keysers et al. IEEE PAMI 2007
K-NN with non-linear deformation (P2DHMDM)	shiftable edges	0.52	Keysers et al. IEEE PAMI 2007
K-NN, Tangent Distance	subsampling to 16x16 pixels	1.1	LeCun et al. 1998
K-NN, shape context matching	shape context feature extraction	0.63	Belongie et al. IEEE PAMI 2002
Boosted Stumps			
boosted stumps	none	7.7	Kegl et al., ICML 2009
products of boosted stumps (3 terms)	none	1.26	Kegl et al., ICML 2009
boosted trees (17 leaves)	none	1.53	Kegl et al., ICML 2009
stumps on Haar features	Haar features	1.02	Kegl et al., ICML 2009
product of stumps on Haar f.	Haar features	0.87	Kegl et al., ICML 2009
Non-Linear Classifiers			
40 PCA + quadratic classifier	none	3.3	LeCun et al. 1998
1000 RBF + linear classifier	none	3.6	LeCun et al. 1998
SVMs			

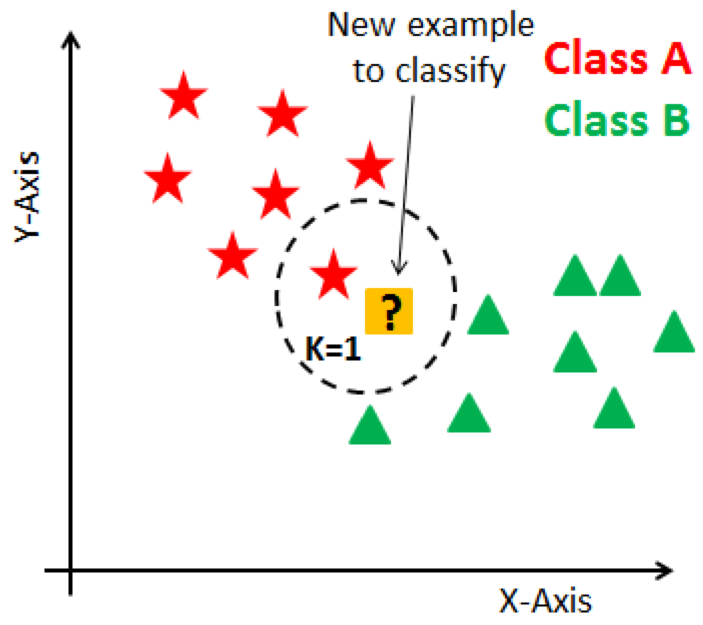
CLASSIFIER	PREPROCESSING	TEST ERROR RATE (%)	Reference
Linear Classifiers			
linear classifier (1-layer NN)	none	12.0	LeCun et al. 1998
linear classifier (1-layer NN)	deskewing	8.4	LeCun et al. 1998
pairwise linear classifier	deskewing	7.6	LeCun et al. 1998
K-Nearest Neighbors			
K-nearest-neighbors, Euclidean (L2)	none	5.0	LeCun et al. 1998
K-nearest-neighbors, Euclidean (L2)	none	3.09	Kenneth Wilder, U. Chicago
K-nearest-neighbors, L3	none	2.83	Kenneth Wilder, U. Chicago
K-nearest-neighbors, Euclidean (L2)	deskewing	2.4	LeCun et al. 1998
K-nearest-neighbors, Euclidean (L2)	deskewing, noise removal, blurring	1.80	Kenneth Wilder, U. Chicago
K-nearest-neighbors, L3	deskewing, noise removal, blurring	1.73	Kenneth Wilder, U. Chicago
K-nearest-neighbors, L3	deskewing, noise removal, blurring, 1 pixel shift	1.33	Kenneth Wilder, U. Chicago
K-nearest-neighbors, L3	deskewing, noise removal, blurring, 2 pixel shift	1.22	Kenneth Wilder, U. Chicago
K-NN with non-linear deformation (IDM)	shiftable edges	0.54	Keysers et al. IEEE PAMI 2007
K-NN with non-linear deformation (P2DHMDM)	shiftable edges	0.52	Keysers et al. IEEE PAMI 2007
K-NN, Tangent Distance	subsampling to 16x16 pixels	1.1	LeCun et al. 1998
K-NN, shape context matching	shape context feature extraction	0.63	Belongie et al. IEEE PAMI 2002
Boosted Stumps			
boosted stumps	none	7.7	Kegl et al., ICML 2009
products of boosted stumps (3 terms)	none	1.26	Kegl et al., ICML 2009
boosted trees (17 leaves)	none	1.53	Kegl et al., ICML 2009
stumps on Haar features	Haar features	1.02	Kegl et al., ICML 2009
product of stumps on Haar f.	Haar features	0.87	Kegl et al., ICML 2009
Non-Linear Classifiers			
40 PCA + quadratic classifier	none	3.3	LeCun et al. 1998
1000 RBF + linear classifier	none	3.6	LeCun et al. 1998
SVMs			

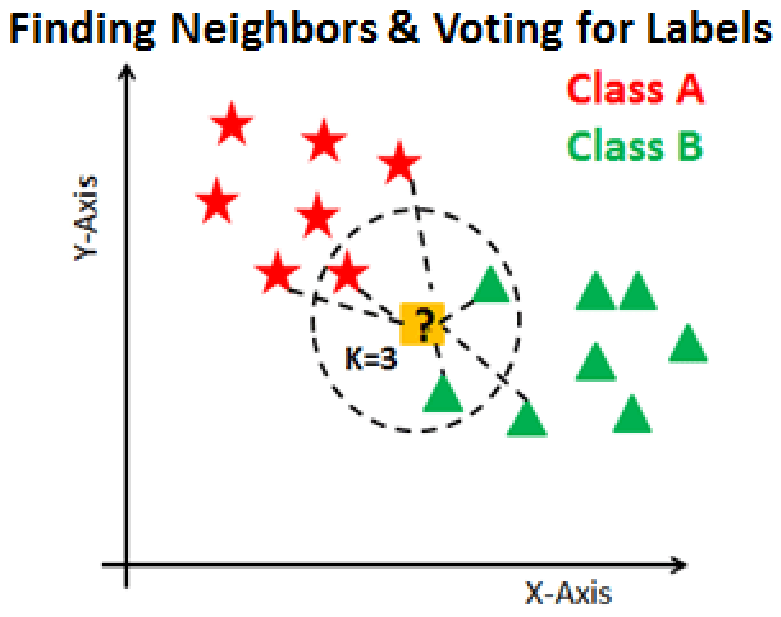
K-NN, Tangent Distance

- Was the MNIST problem “solved” in 1998?
- Is the solution in 1998 valid?

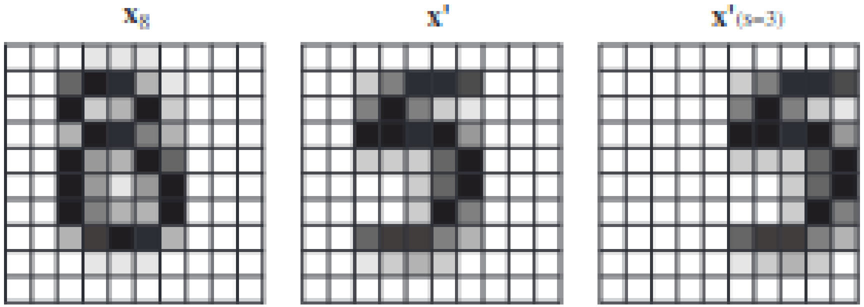
Invariance

Nearest-Neighbor Classification



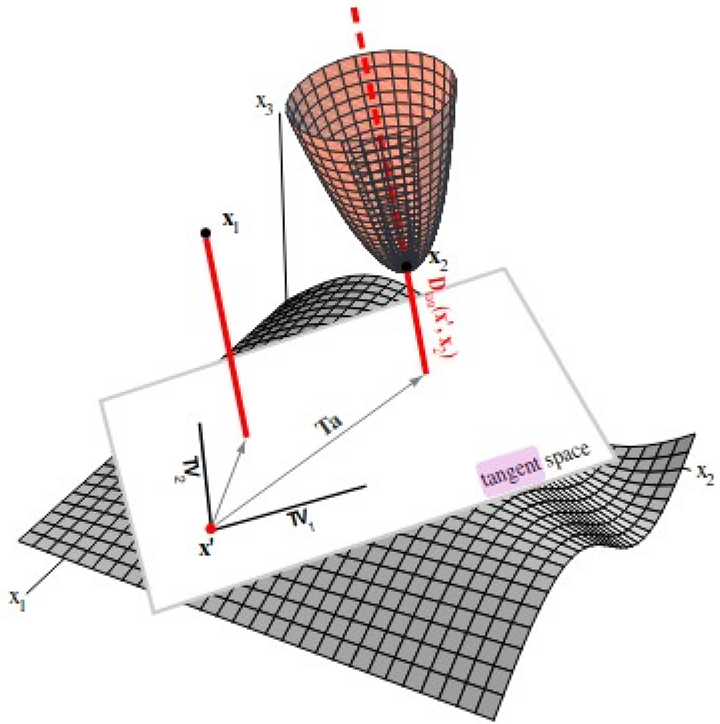


k-NN on MNIST



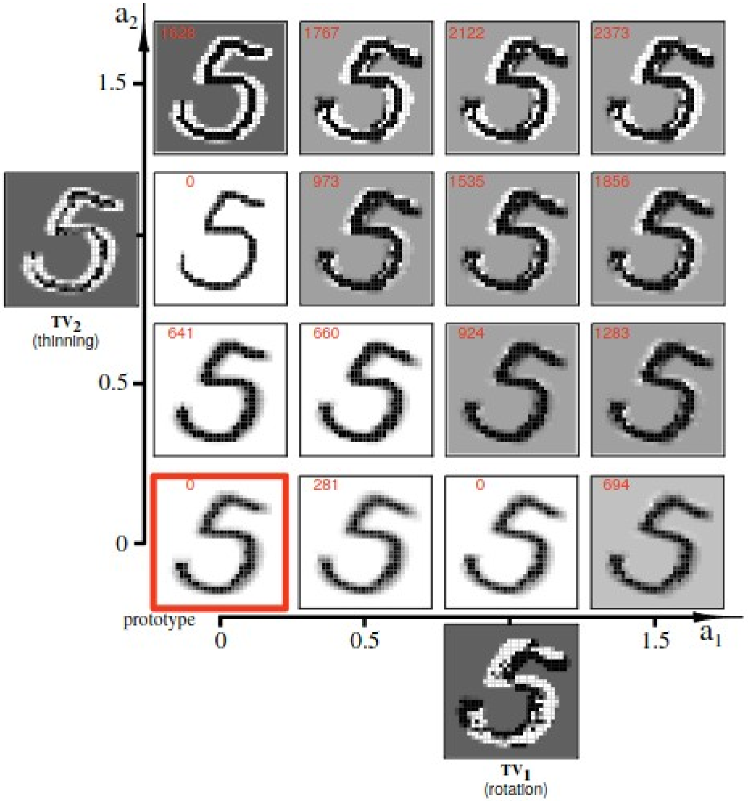
- What would be the problem of using Euclidean distance?

Tangent Distance



$$D_{tan}(\mathbf{x}', \mathbf{x}) = \min_{\mathbf{a}} [||(\mathbf{x}' + \mathbf{Ta}) - \mathbf{x}||]$$

Tangential Vector



K-NN with Tangential Distance in MNIST

K-NN, Tangent Distance

- Was the MNIST problem “solved” in 1998?

- Is the solution in 1998 valid?

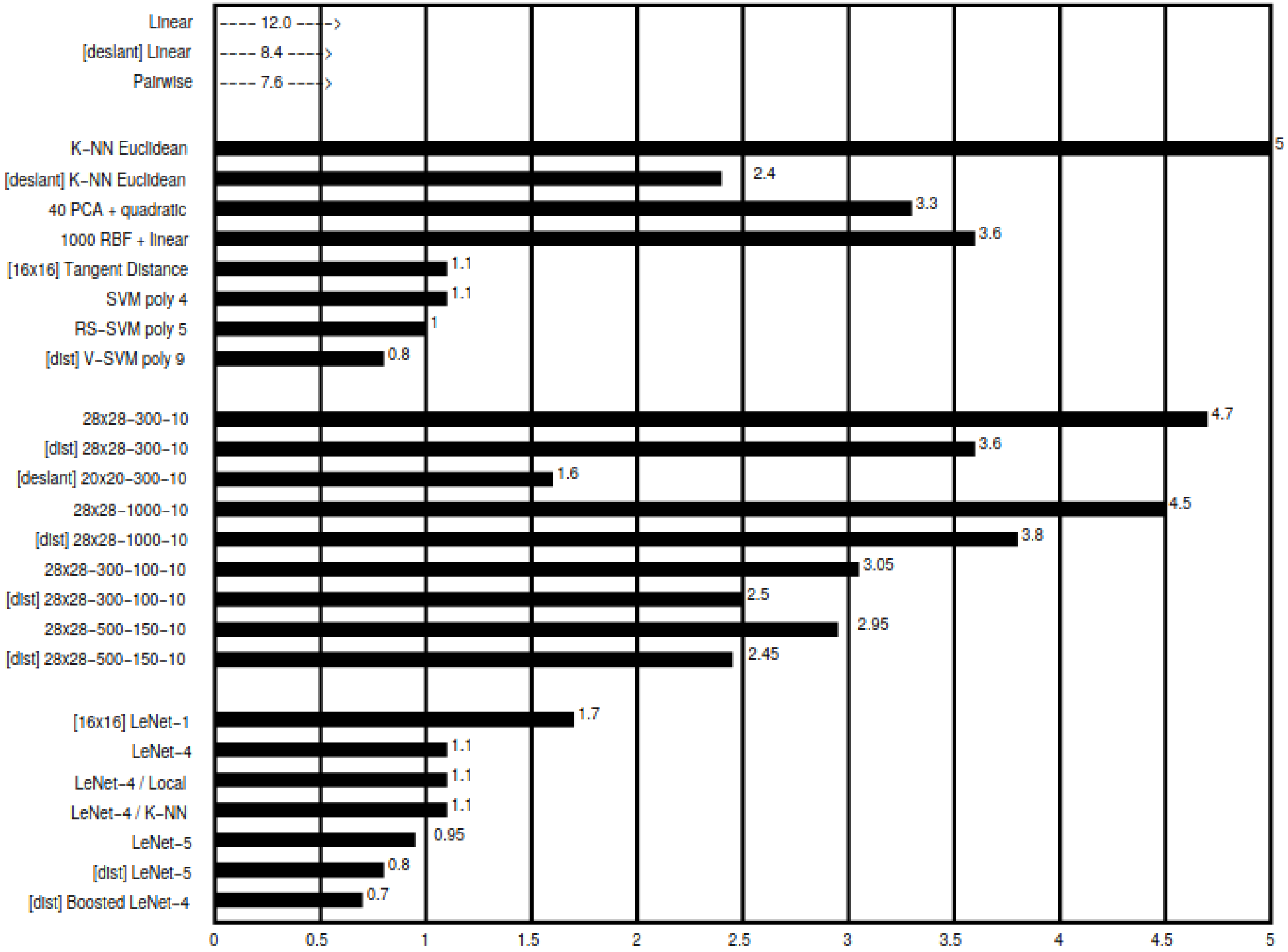


Fig. 9. Error rate on the test set (%) for various classification methods. [deslant] indicates that the classifier was trained and tested on the deslanted version of the database. [dist] indicates that the training set was augmented with artificially distorted examples. [16x16] indicates that the system used the 16x16 pixel images. The uncertainty in the quoted error rates is about 0.1%.

What does matter in the Algorithm?

- Accuracy
- Latency and Throughput
- Training time and inference time
- Memory requirements

What’s up for today?

- A kind archeology tour
- **Examples in Meteorology**
- Examples in Eye Tracking

Meteorology

Deep Learning ~~Winter~~ is coming...

Segmenting and Tracking Extreme Climate Events using Neural Networks

Mayur Mudigonda[*1], **Sookyung Kim**[*2], **Ankur Mahesh**[*1], **Samira Kahou**
Karthik Kashinath[4], **Dean Williams**[2], **Vincent Michalski**[5], **Travis O'Brien**[4] **and M**
University of California Berkeley[1], Lawrence Livermore National Lab[2], Microsoft R
Lawrence Berkeley National Lab[4], Montreal Institute for Learning Algorithms
mudigonda@berkeley.edu, kim79@llnl.gov, ankur.mahesh@berkeley.
samira.ebrahimi.kahou@gmail.com, kkashinath@lbl.gov, williams13@ll
michalskivincent@gmail.com, taobrien@lbl.gov, prabhat@lbl.go

Abstract

Predicting extreme weather events in a warming world is one of the most pressin
and challenging problems that humanity faces today. Deep learning and advance

Deep Learning for Precipitation Nowcasting: A Benchmark and A New Model

Xingjian Shi, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
{xshiab,zgaoag,lelausen,hwangaz,dyyeung}@cse.ust.hk

Wai-kin Wong, Wang-chun Woo
Hong Kong Observatory
Hong Kong, China
{wkwong,wcwoo}@hko.gov.hk

Abstract

With the goal of making high-resolution forecasts of regional rainfall, precipitation nowcasting has become an important and fundamental technology underlying various public services ranging from rainstorm warnings to flight safety. Recently,

Deep Learning ~~Winter~~ is coming...

Google AI Blog

The latest news from Google AI

A Neural Weather Model for Eight-Hour Precipitation Forecasting

Wednesday, March 25, 2020

Posted by Nal Kalchbrenner and Casper Sønderby, Research Scientists, Google Research, Amsterdam

Predicting weather from minutes to weeks ahead with high accuracy is a fundamental scientific challenge that can have a wide ranging impact on many aspects of society. Current forecasts employed by many meteorological agencies are based on physical models of the atmosphere that, despite improving substantially over the preceding decades, are inherently constrained by their computational requirements and are sensitive to approximations of the physical laws that govern them. An alternative approach to weather prediction that is able to overcome some of these constraints uses deep neural networks (DNNs): instead of encoding explicit physical laws, DNNs discover patterns in the data and learn complex transformations from inputs to the desired outputs using parallel computation on powerful specialized hardware such as GPUs and TPUs.

Building on our previous research into precipitation nowcasting, we present "MetNet: A Neural Weather Model for Precipitation Forecasting," a DNN capable of predicting future precipitation at 1 km resolution over 2 minute intervals at timescales up to 8 hours into the future. MetNet outperforms the current state-of-the-art physics-based model in use by NOAA for prediction times up to 7-8 hours ahead and makes a prediction over the entire US in a matter of seconds as opposed to an hour. The inputs to the network are sourced automatically from radar stations and satellite networks without the need for human annotation. The model output is a probability distribution that we use to infer the most likely precipitation rates with associated uncertainties at each geographical region. The figure below provides an example of the network's predictions over the continental United States.

Using Machine Learning to "Nowcast" Precipitation in High Resolution

Monday, January 13, 2020

Posted by Jason Hickey, Senior Software Engineer, Google Research

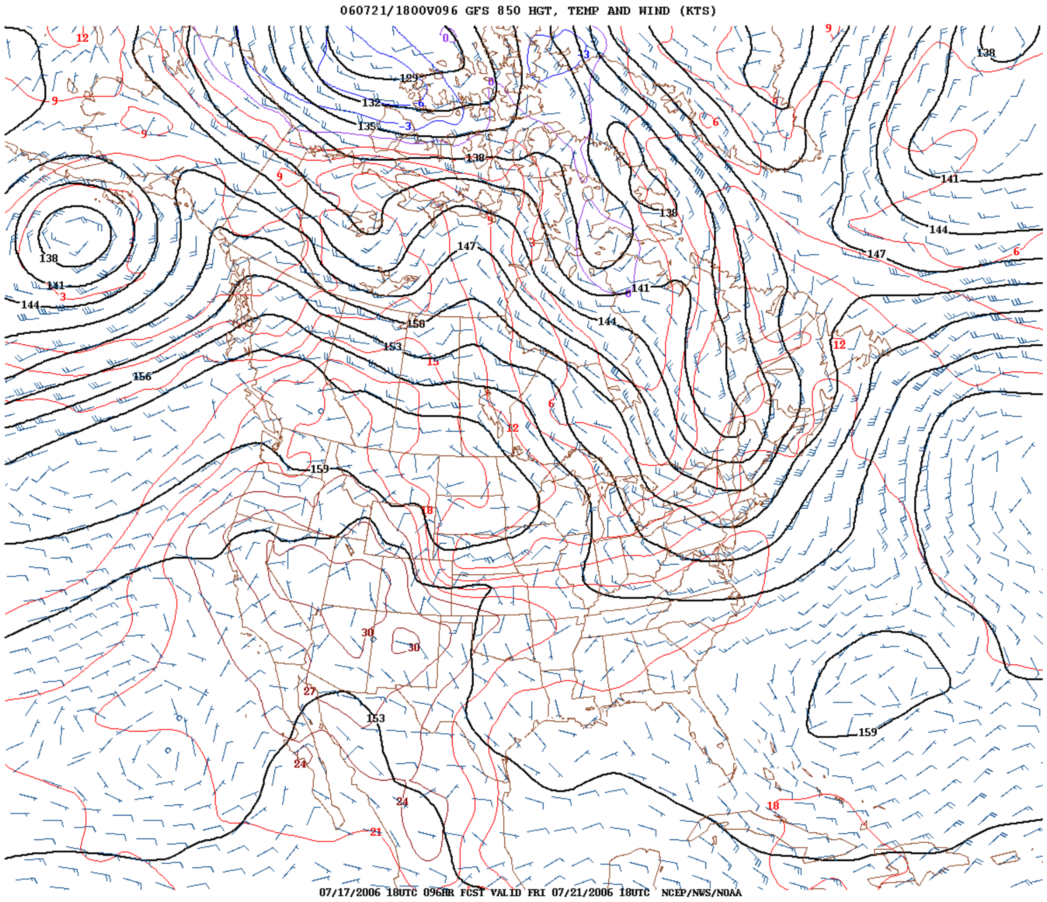
The weather can affect a person's daily routine in both mundane and serious ways, and the precision of forecasting can strongly influence how they deal with it. Weather predictions can inform people about whether they should take a different route to work, if they should reschedule the picnic planned for the weekend, or even if they need to evacuate their homes due to an approaching storm. But making accurate weather predictions can be particularly challenging for localized storms or events that evolve on hourly timescales, such as thunderstorms.

In "Machine Learning for Precipitation Nowcasting from Radar Images," we are presenting new research into the development of machine learning models for precipitation forecasting that addresses this challenge by making highly localized "physics-free" predictions that apply to the immediate future. A significant advantage of machine learning is that inference is computationally cheap given an already-trained model, allowing forecasts that are nearly instantaneous and in the native high resolution of the input data. This *precipitation nowcasting*, which focuses on 0-6 hour forecasts, can generate forecasts that have a 1km resolution with a total latency of just 5-10 minutes, including data collection delays, outperforming traditional models, even at these early stages of development.

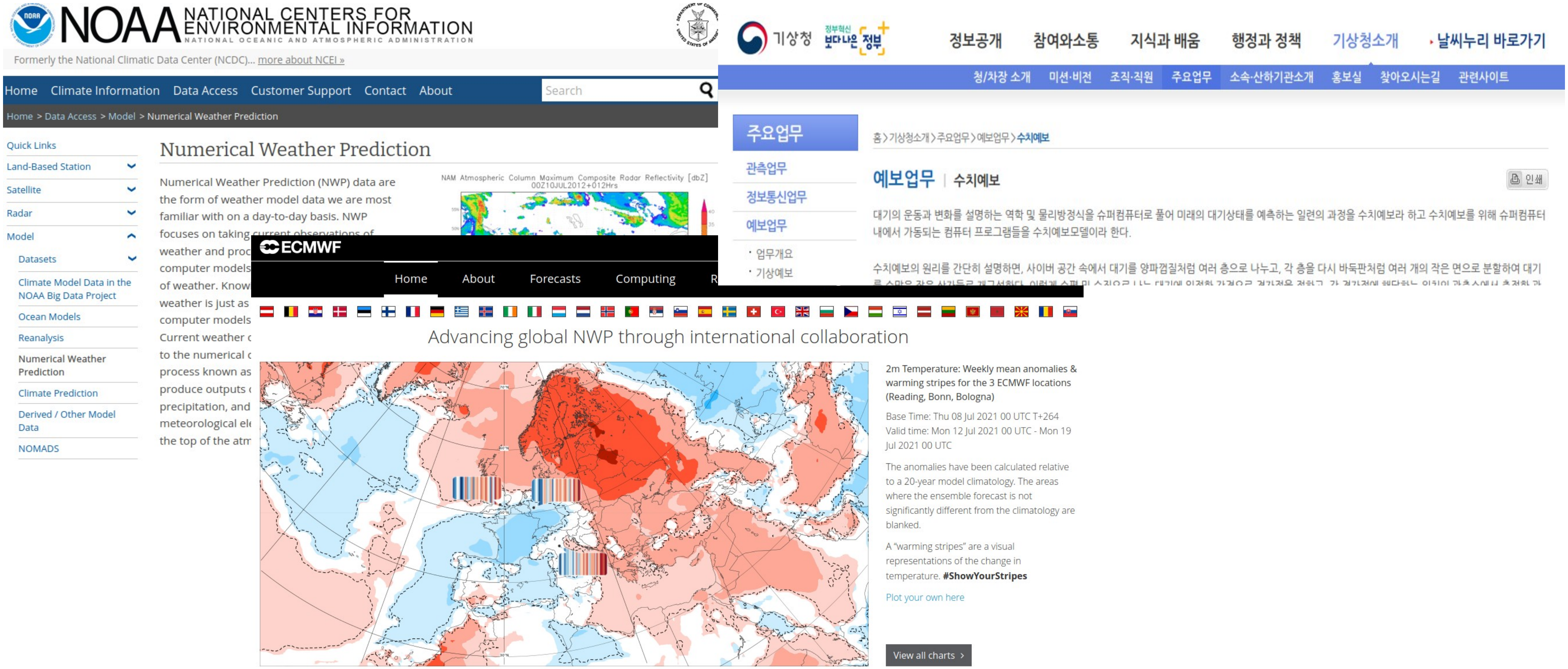
Moving Beyond Traditional Weather Forecasting

Weather agencies around the world have extensive monitoring facilities. For example, Doppler radar measures precipitation in real-time, weather satellites provide multispectral imaging, ground stations measure wind and precipitation directly, etc. The figure below, which compares false-color composite radar imaging of precipitation over the continental US to cloud cover imaged by geosynchronous satellites, illustrates the need for multi-source weather information. The existence of rain is related to, but not perfectly correlated with, the existence of clouds, so inferring precipitation from satellite images alone is challenging.

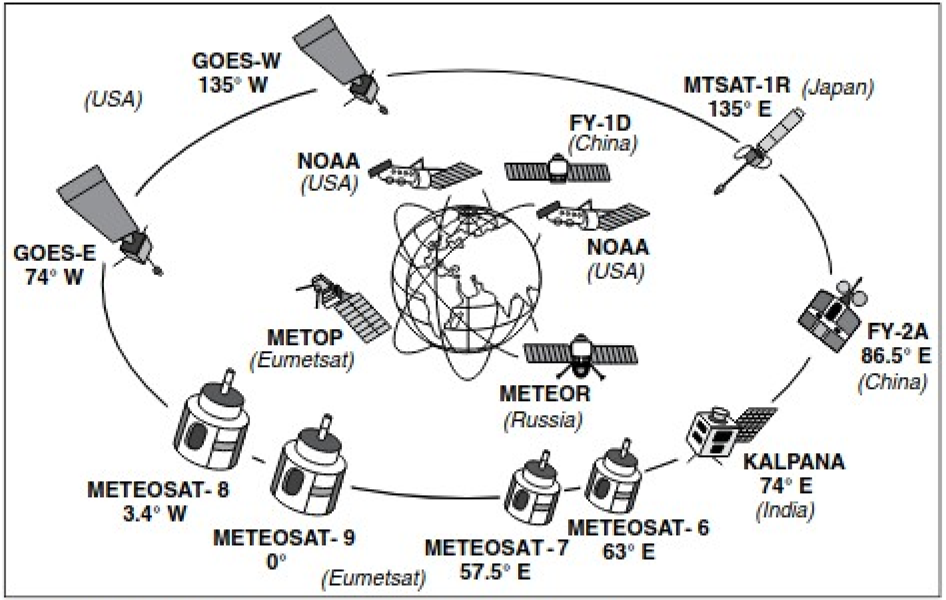
Numerical Weather Forecasting



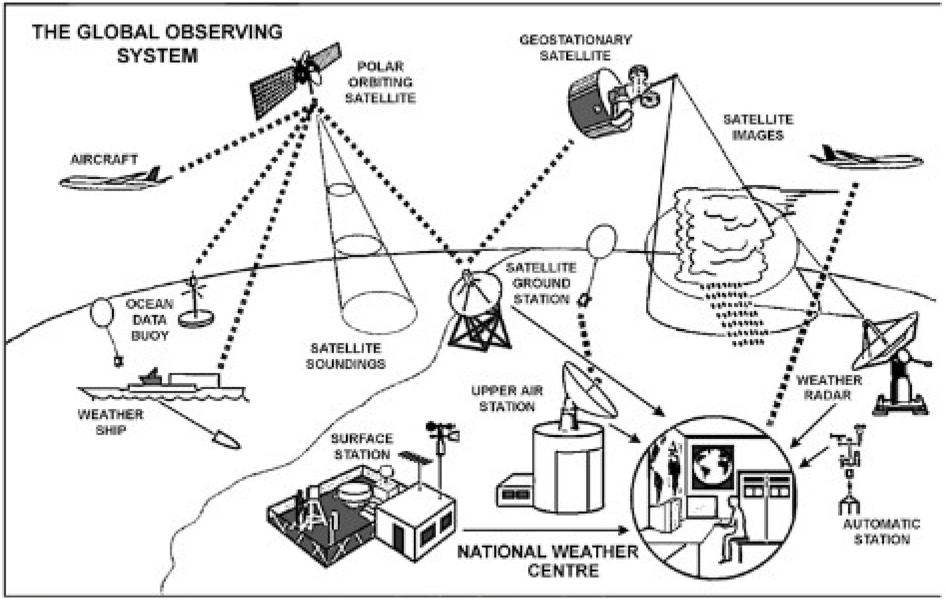
Weather Agency



Observation



The operational system of meteorological satellites in 2007. (From EUMETSAT)



The various components of the Global Observing System. (After a WMO image)

Data Assimilation

3D-Var [edit]

$$J(\mathbf{x}) = (\mathbf{x} - \mathbf{x}_b)^{\mathrm{T}} \mathbf{B}^{-1} (\mathbf{x} - \mathbf{x}_b) + (\mathbf{y} - H[\mathbf{x}])^{\mathrm{T}} \mathbf{R}^{-1} (\mathbf{y} - H[\mathbf{x}]),$$

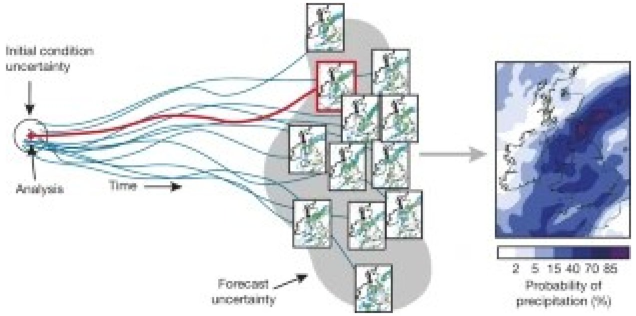
where $\mathbf{B}$ denotes the background error covariance, $\mathbf{R}$ the observational error covariance.

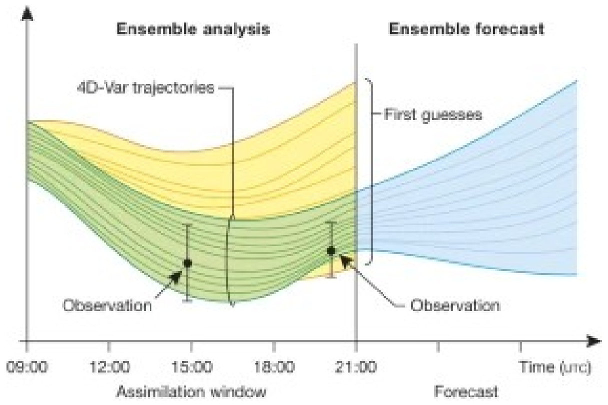
$$\nabla J(\mathbf{x}) = 2\mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}_b) - 2H^T \mathbf{R}^{-1}(\mathbf{y} - H[\mathbf{x}])$$

4D-Var [edit]

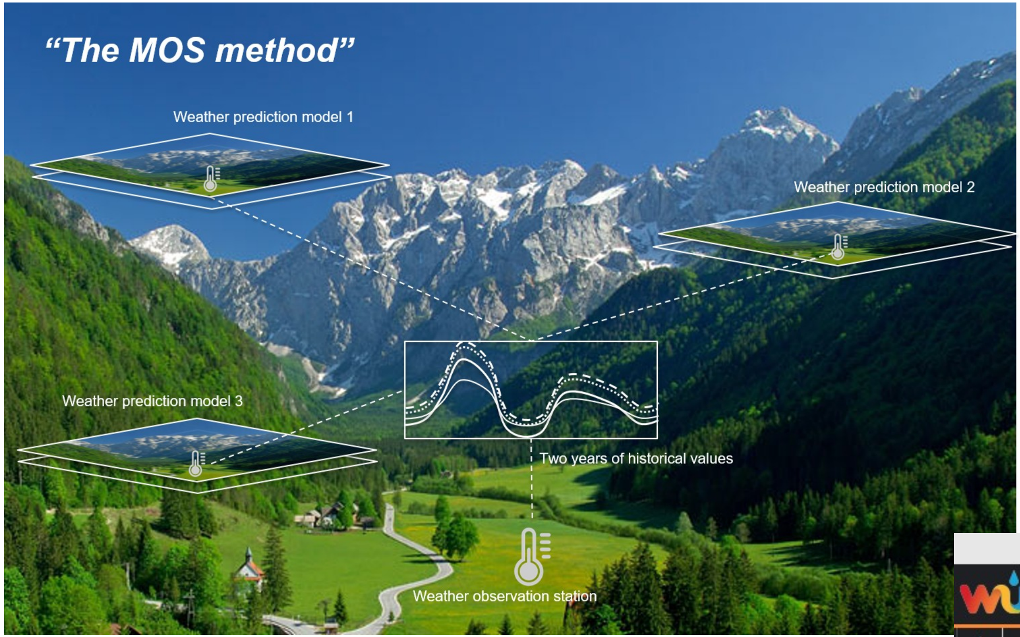
$$J(\mathbf{x}) = (\mathbf{x} - \mathbf{x}_b)^{\mathrm{T}} \mathbf{B}^{-1} (\mathbf{x} - \mathbf{x}_b) + \sum_{i=0}^{n} (\mathbf{y}_i - H_i[\mathbf{x}_i])^{\mathrm{T}} \mathbf{R}_i^{-1} (\mathbf{y}_i - H_i[\mathbf{x}_i])$$

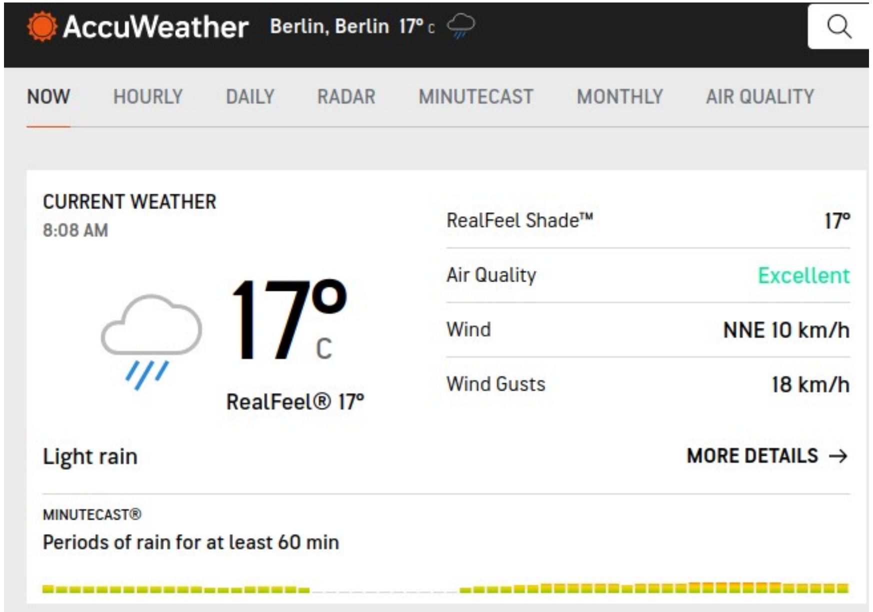
provided that H is a linear operator (matrix).

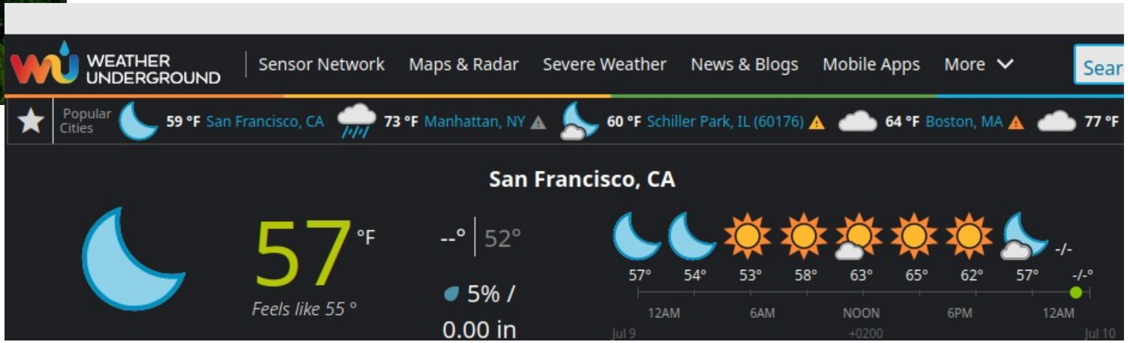




MOS (Model Output Statistics)







Deep Learning ~~Winter~~ is coming...

March 3, 2021

Numerical weather prediction (NWP) is a mainstay of supercomputing. Some of the first applications of the first supercomputers dealt with climate modelin and even to this day, the largest climate models are heavily constrained by th scale of the supercomputers that run them. While some wait for the exascale era – and beyond – to brute force punishingly accurate and complex climate models into existence, others are looking for a deep learning-powered shortc to the same results. In a paper for *Philosophical Transactions of the Royal Society*, eight researchers from the Jülich Supercomputing Center explored whether deep learning could ever actually beat numerical weather prediction at its own game – and if so, how and when that might happen.

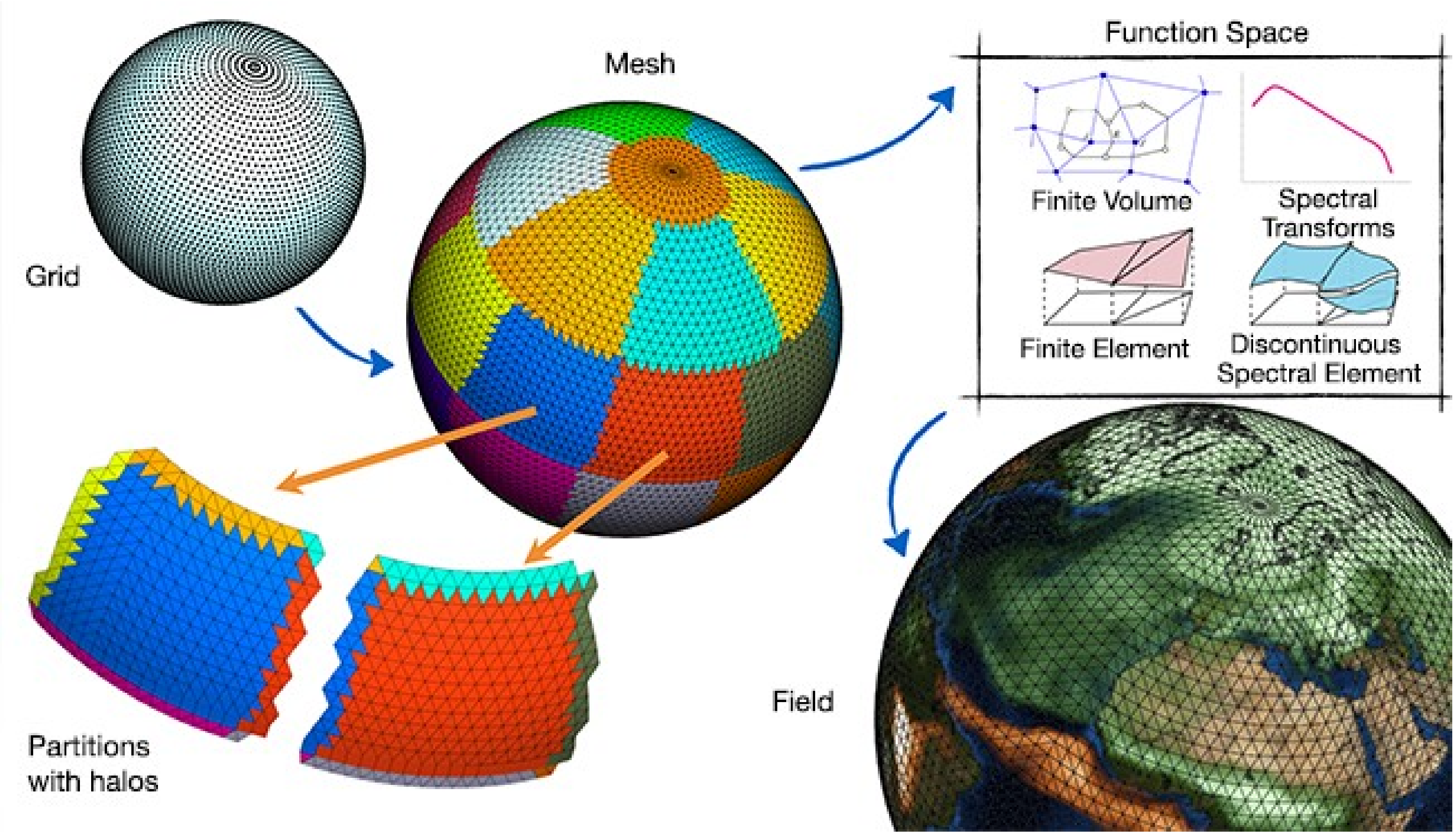
Status quo

The weather and climate supercomputing community is no stranger to deep learning, but it has hitherto mostly been used to augment NWP approaches (e.g. in resolving post-processing issues). These modelers, however, are reticent to incorporate deep learning in more meaningful capacities.

"[There] are still reservations about DL in this community," the authors write. "Two core arguments in this regard are the lack of explainability of deep [neural networks] and the lack of physical constraints. Furthermore, some

"[There] are still reservations about DL in this community," the authors write. "Two core arguments in this regard are the lack of explainability of deep [neural networks] and the lack of physical constraints. Furthermore, some scepticism prevails due to the fact that researchers have experimented with rather simple [neural networks] which were clearly unsuited to capture the complexity of meteorological data and feedback processes, but then extrapolate these results to discredit any [neural network] application including the much more powerful [deep learning] systems."

Spatial Grid and Scalability



Can Deep Learning Replace Numerical Weather Forecasting?

What does matter in the Algorithm?

- Accuracy: **BENCHMARK DATASET**
- Latency and Throughput
- Training time and inference time
- Memory requirements

Parameter/Model Optimization

GLOBAL SPATIAL GRID

Model Drift

Numerical Weather Forecasting

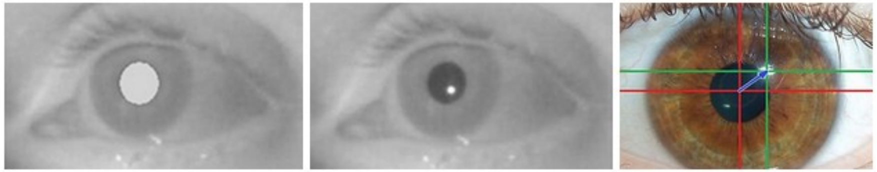
- Is the weather forecasting problem “solved” in 2021?

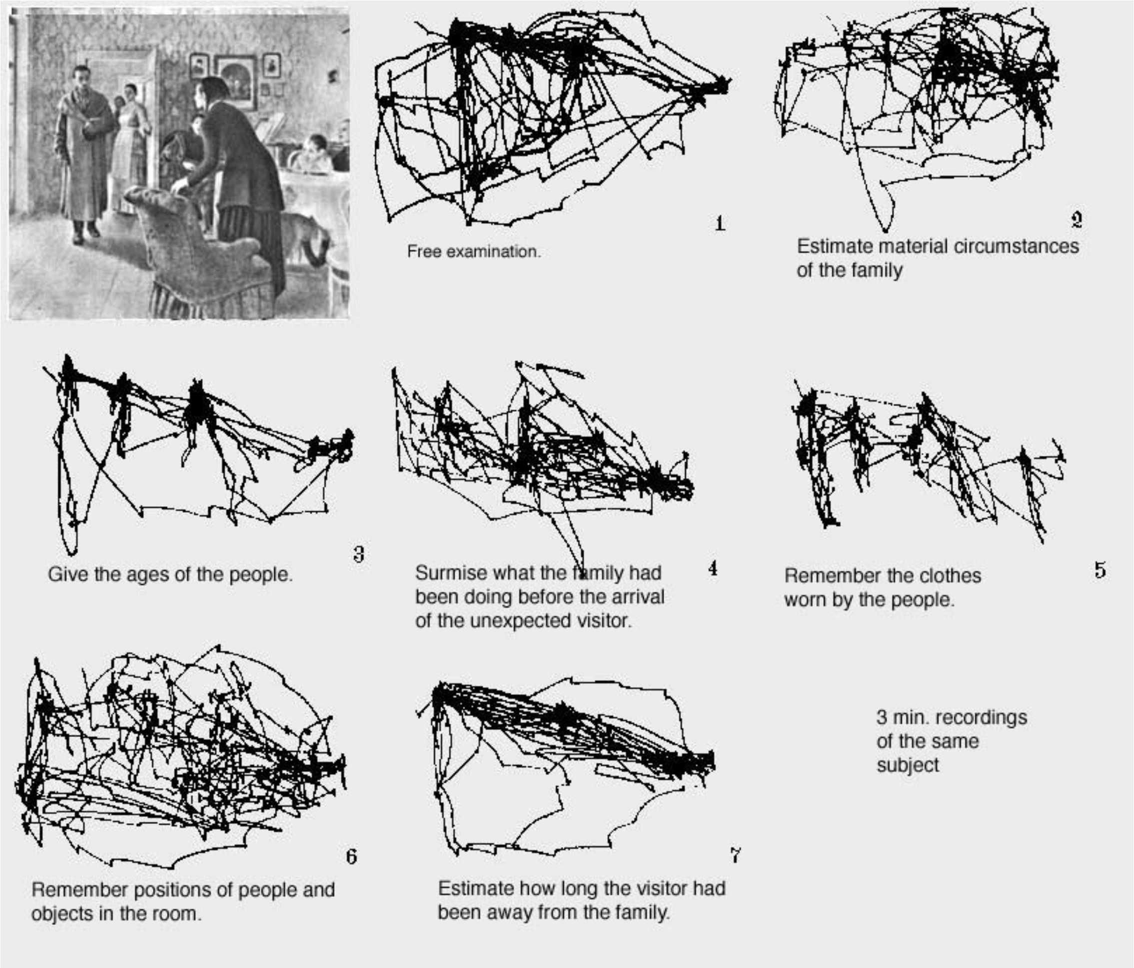
- Will the current solution in 2021 valid in the future?

What’s up for today?

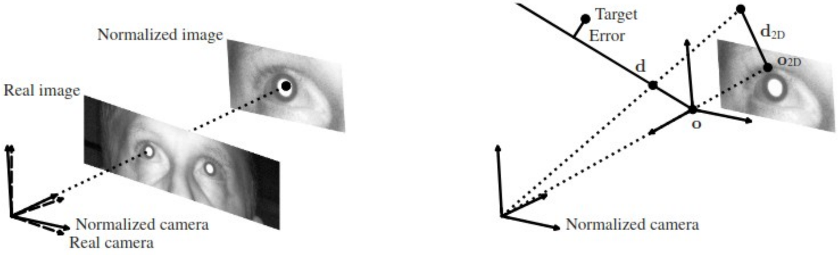
- A kind archeology tour
- Examples in Meteorology
- **Examples in Eye Tracking**

Eye Tracking





Neural Network in Eye Tracking



(a) Image normalization

(b) 3D gaze projection

Figure 2: (a) The image captured by the physical camera is warped into a normalized camera looking directly at the reference point, in this case the persons right eye. (b) The 2D gaze origin and gaze direction are combined with the corrected distance to form a gaze ray in 3D space. The miss distance between the gaze ray and the gaze target is the loss used to train the neural network.

camera coordinate system by $\mathbf{g}(t) = \mathbf{R}^{-1}\hat{\mathbf{g}}(t)$.

3.2. 3D gaze projection

Here we describe how the output from the network is translated into a pair of gaze rays. The network has five outputs. For each eye, it predicts a 2D gaze origin $\mathbf{o}_{2D}$ and a 2D gaze direction $\mathbf{d}_{2D}$. It also generates a distance correction term, c, which is common to both eyes. We assume that the distance from eye to camera is approximately the same for both eyes, and our estimate of that distance will be called ρ. First, given the input image and the eye detections, we find a rough distance ρ_{rough} such that the separation between the eyes becomes 63 mm at that distance, approximately the average human interocular distance [6]. This distance is then

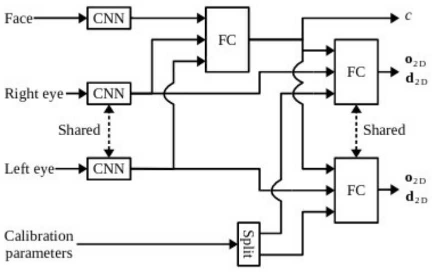


Figure 3: Network architecture.

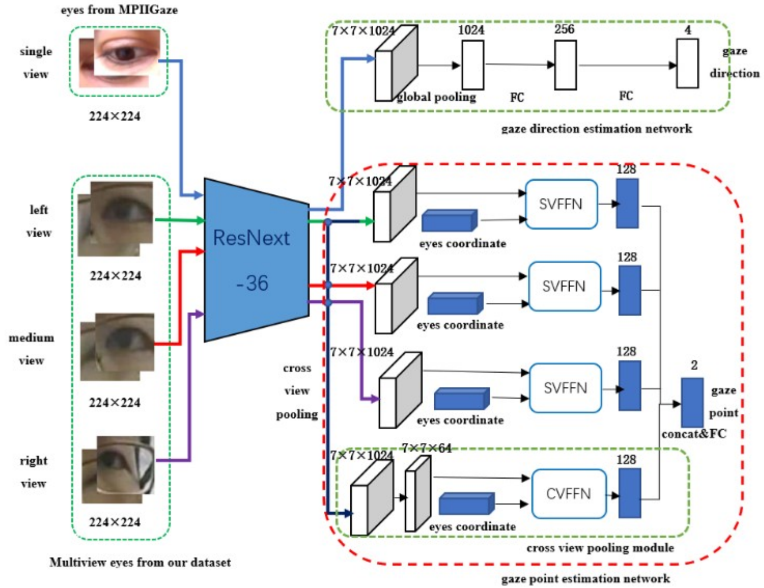
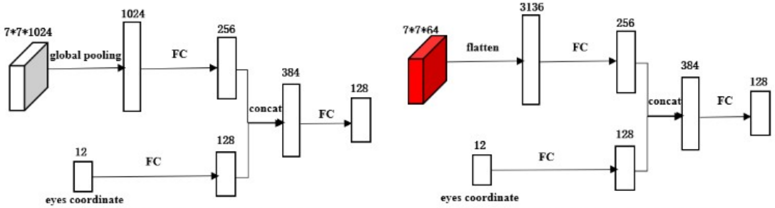


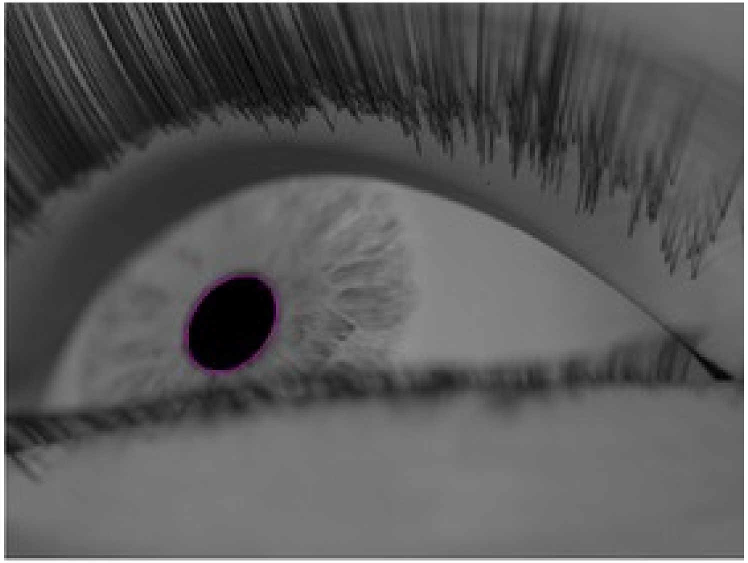
Figure 2: Overall architecture of multiview multitask network for the 3-D gaze direction and 2-D gaze point estimations.

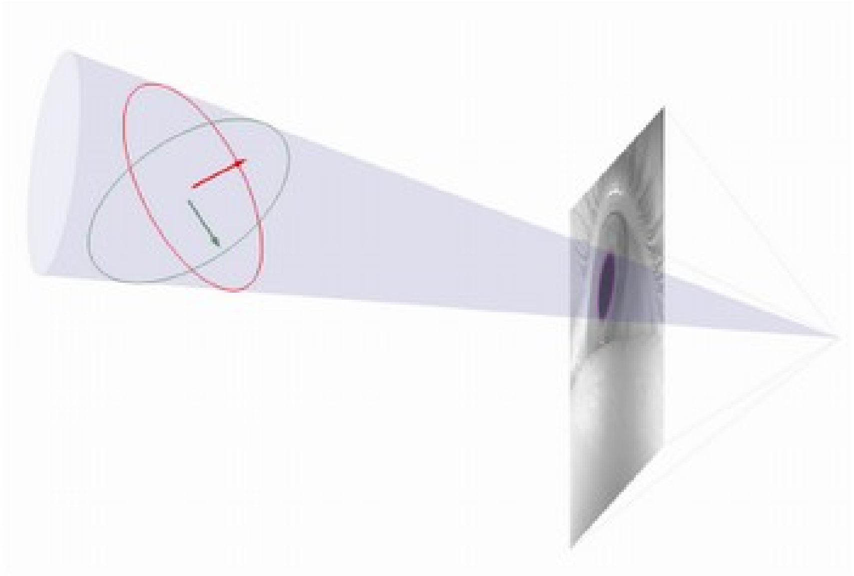


(a) Single View Feature Fusion Network(SVFFN)

(b) Cross View Feature Fusion Network(CVFFN)

3D Eye Model





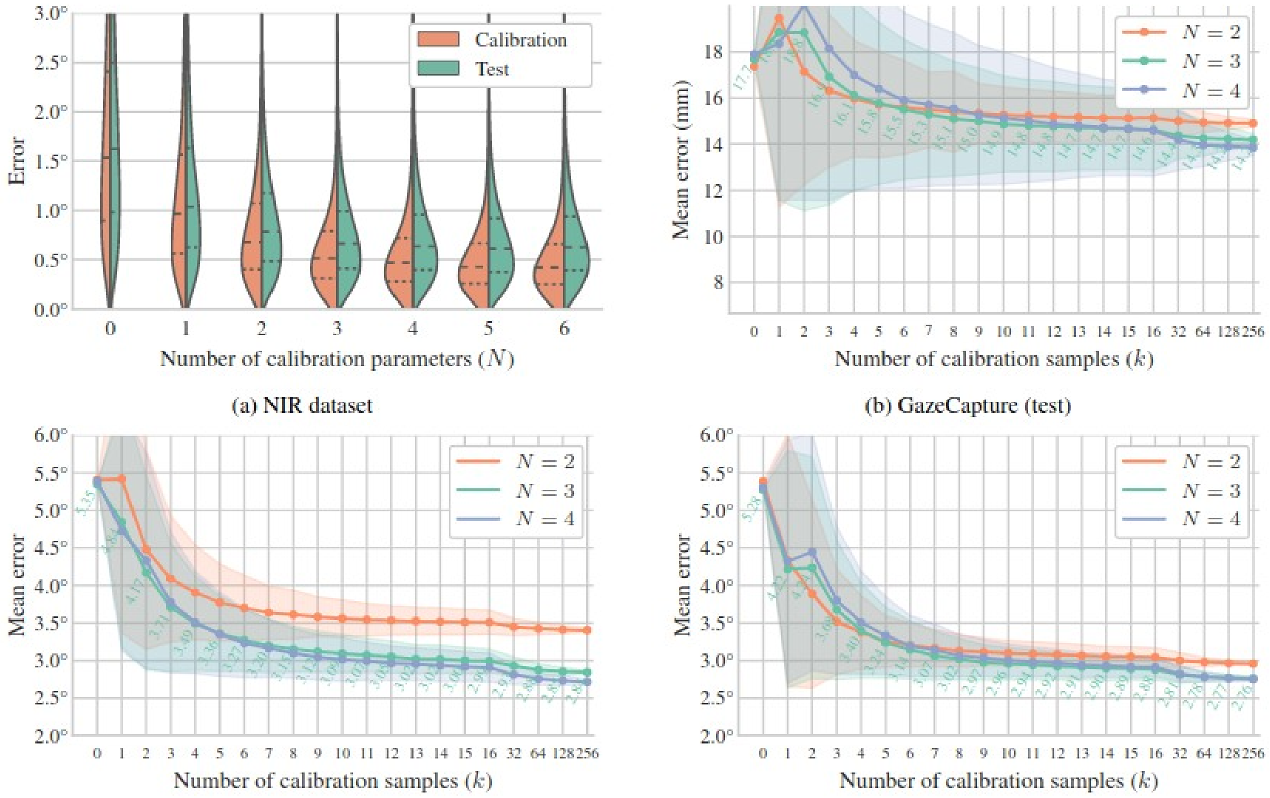
3D Eye Model

	Mean	Std. dev.
Ellipse Unprojection	3.5558	4.2385
Unoptimised Model	2.6890	1.5338
Region Contrast Maximisation	2.2056	0.1629
Edge Distance Minimisation	1.6831	0.3372

	Mean	Std. dev.
Pupil Tracker	3.9301	5.8056
Unoptimised Model	4.4420	4.9157
Region Contrast Maximisation	2.1194	1.1354
Edge Distance Minimisation	2.7474	2.7615

Based on 3D eye model in 2013

Some Gaze Estimation Model based on Neural Network



(a) NIR dataset

(b) GazeCapture (test)

(c) GazeCapture-to-MPIIGaze

(d) Within-MPIIGaze

What does matter in the Algorithm?

- Accuracy
- Latency and Throughput
- Training time and inference time
- Memory requirements

Dataset

Calibration Samples

Gaze Estimation

- Is the Gaze Estimation problem “solved” in 2021?
- Will the current solution in 2021 valid in the future?

Conclusion

- What do you think,

  it will be ***<u>FUNDAMENTAL and CANONICAL</u>***

  in our future?