Using Deep Learning for Long-Term Weather Forecasting

Joshua Driscol
University of Washington

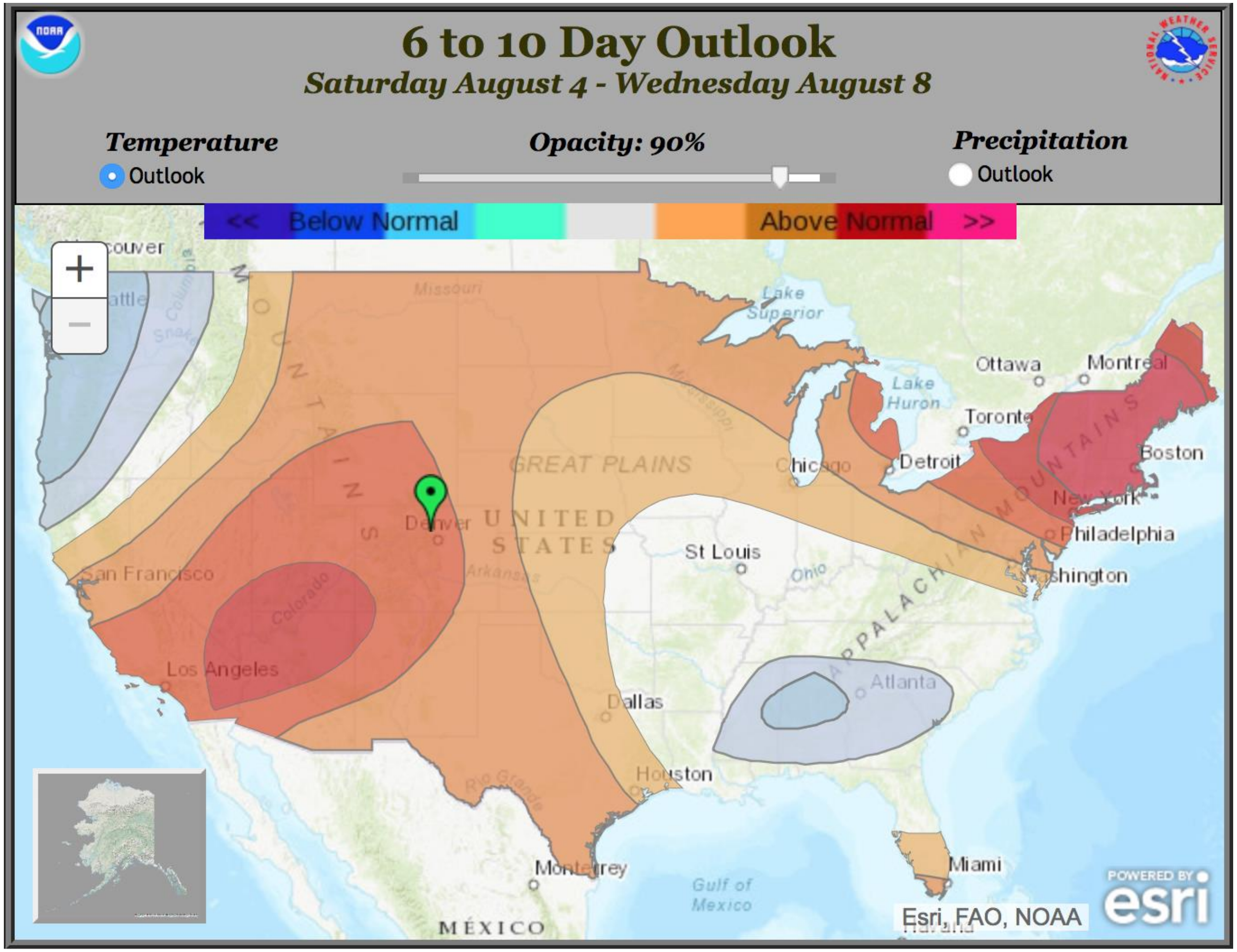
Karen Stengel
Montana State University

06/28/2018

Two main time scales for forecasting:

- *Weather* <= 10 day prediction, or what is currently happening in the atmosphere
- *Climate* is on much longer time scales, and is how we expect the atmosphere to behave
- *Long-term weather*: it would be useful to have accurate predictions for a sub/seasonal timescale

Background

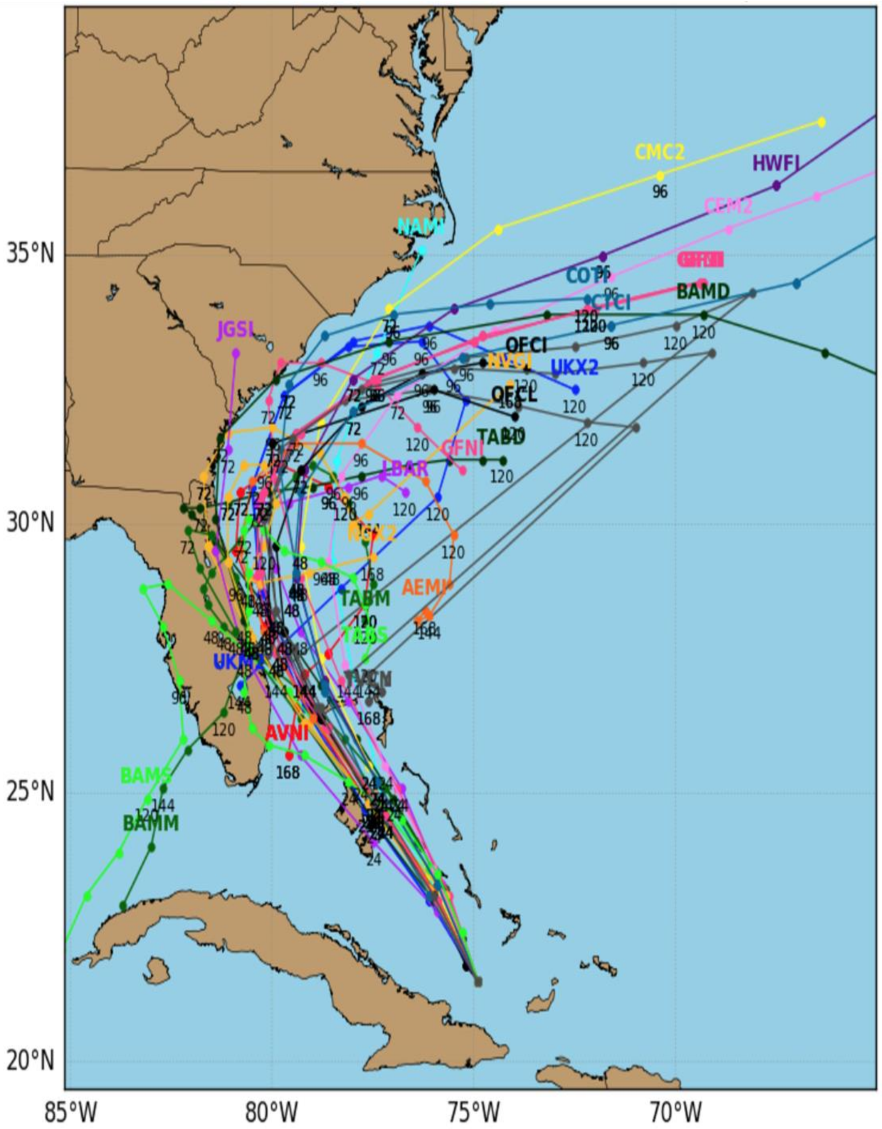


Credit: Climate Prediction Center

Background

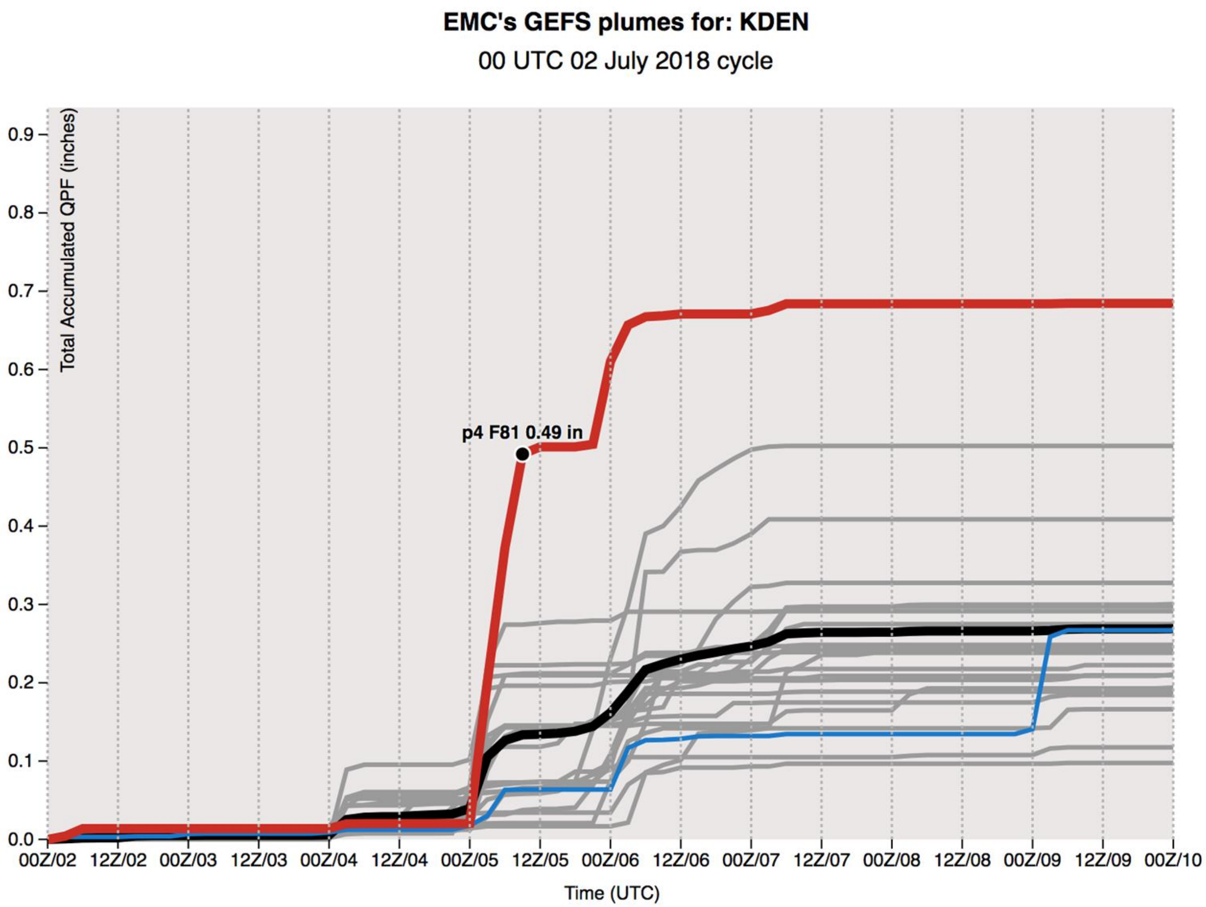
Hurricane Matthew, 2016:

- Over $2 billion damage
- Ended up hitting Florida



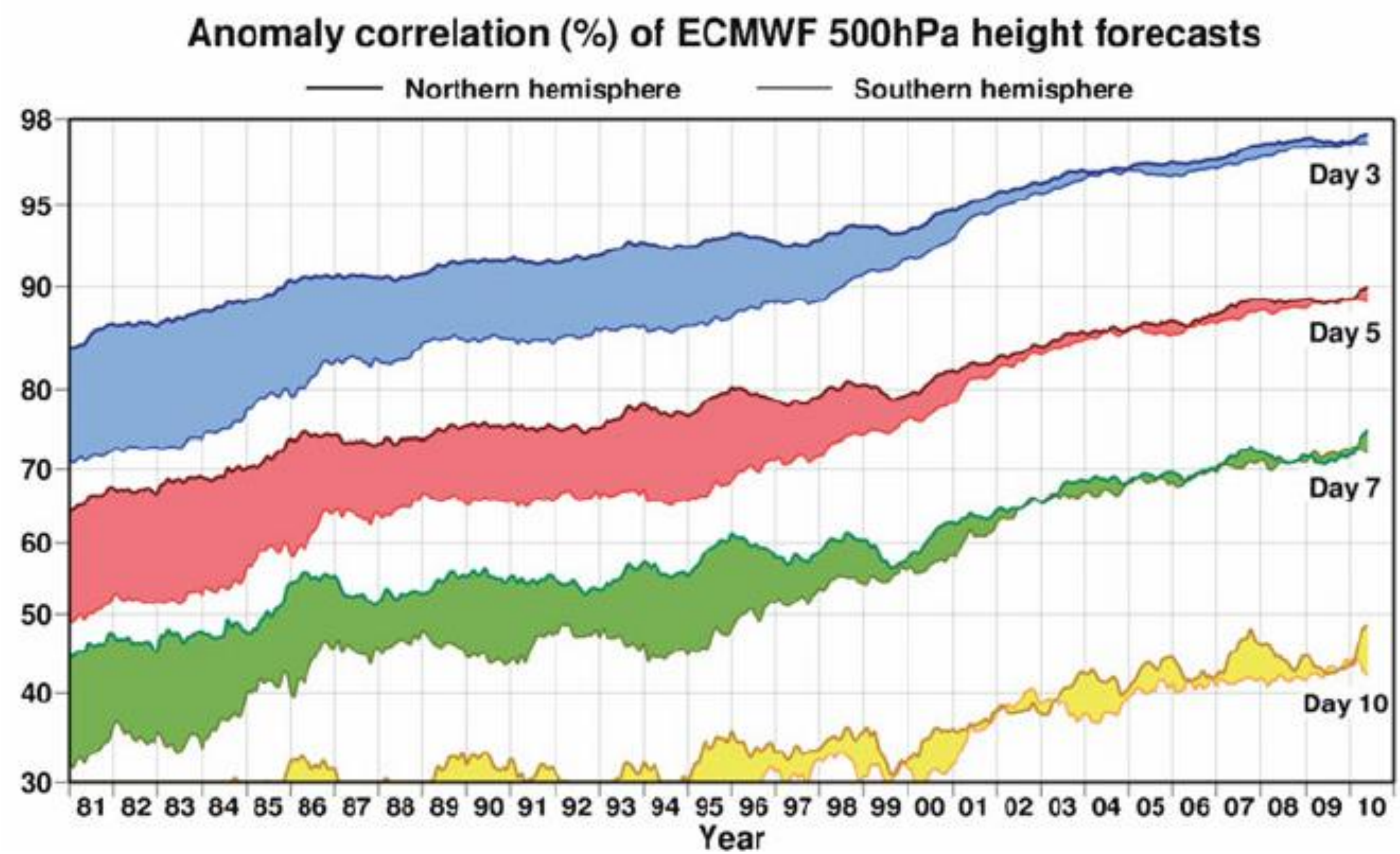
Credit: IBM Weather Company

Not Just



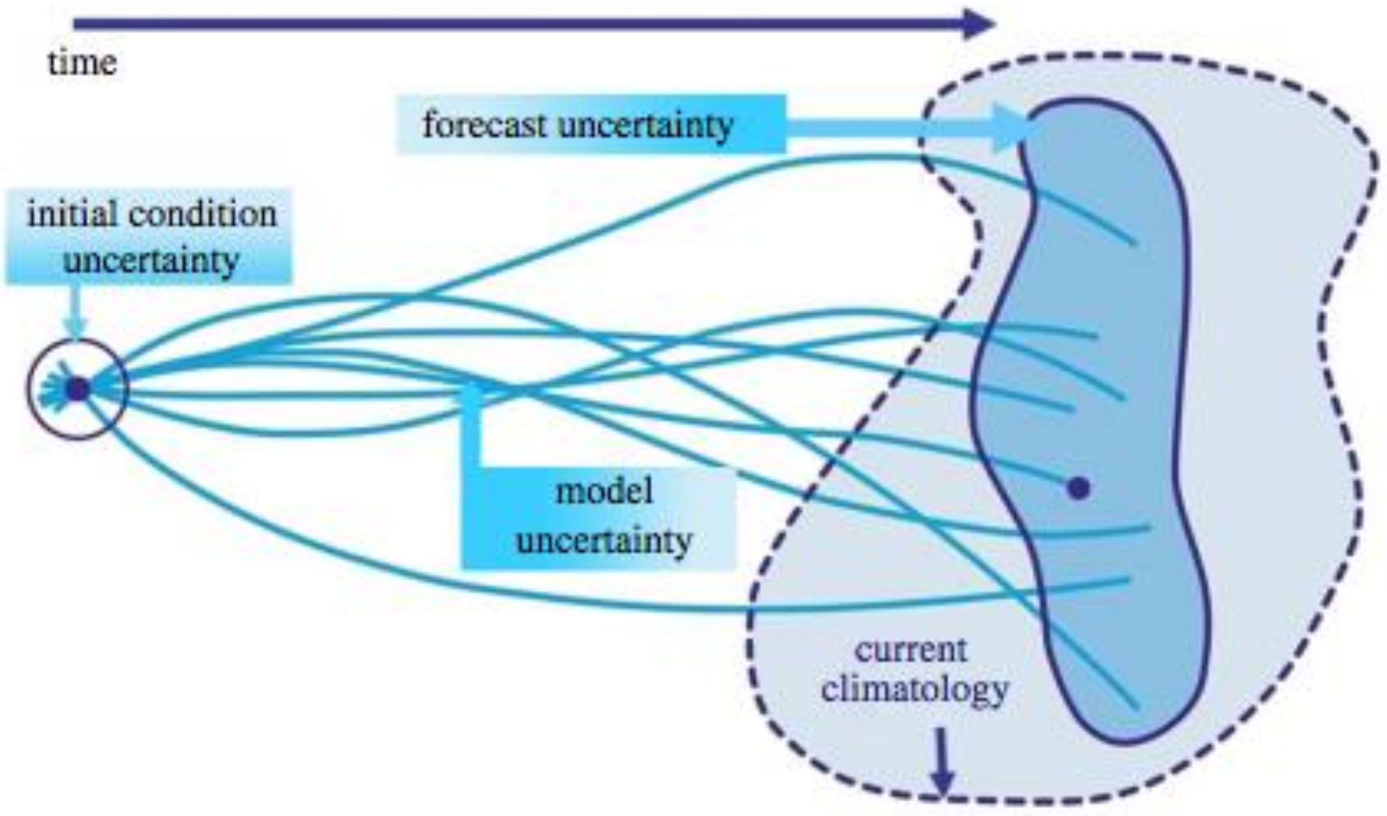
Credit: Environmental Modeling Center (NCEP)

Forecast Skill



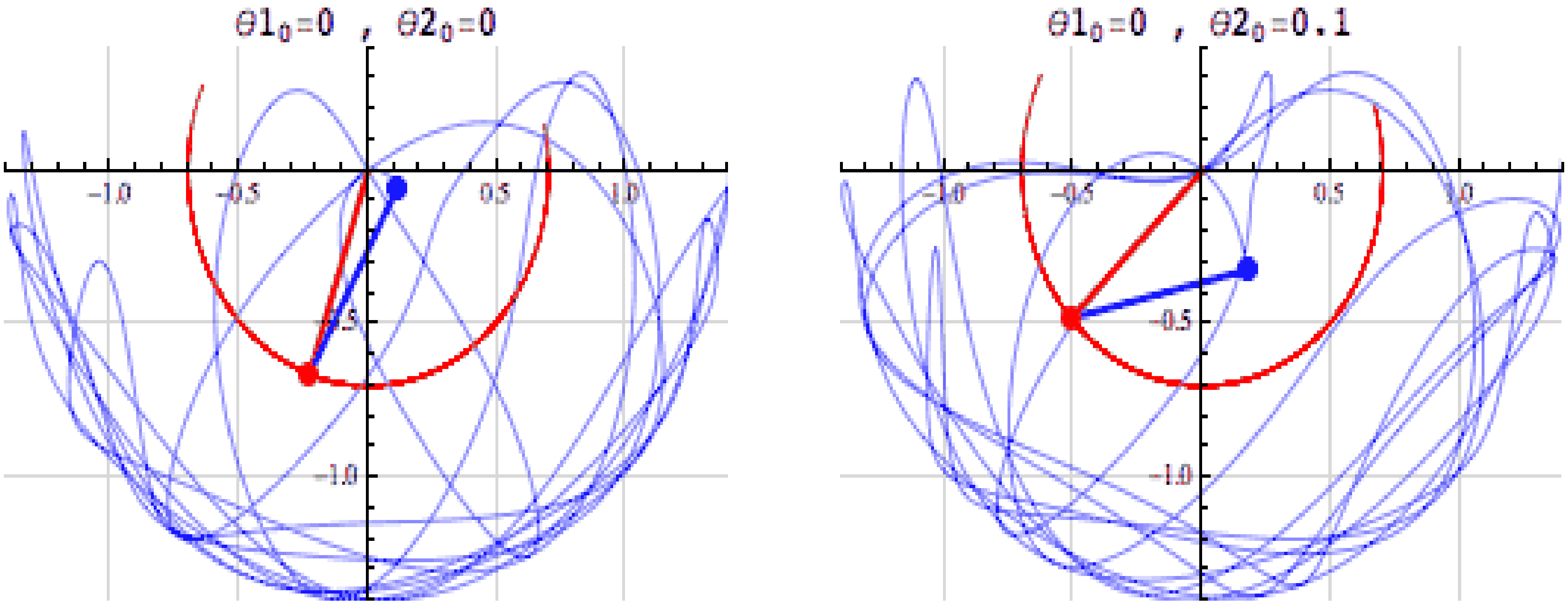
Credit: Kirtman et al, 2011

Mapping Uncertainty



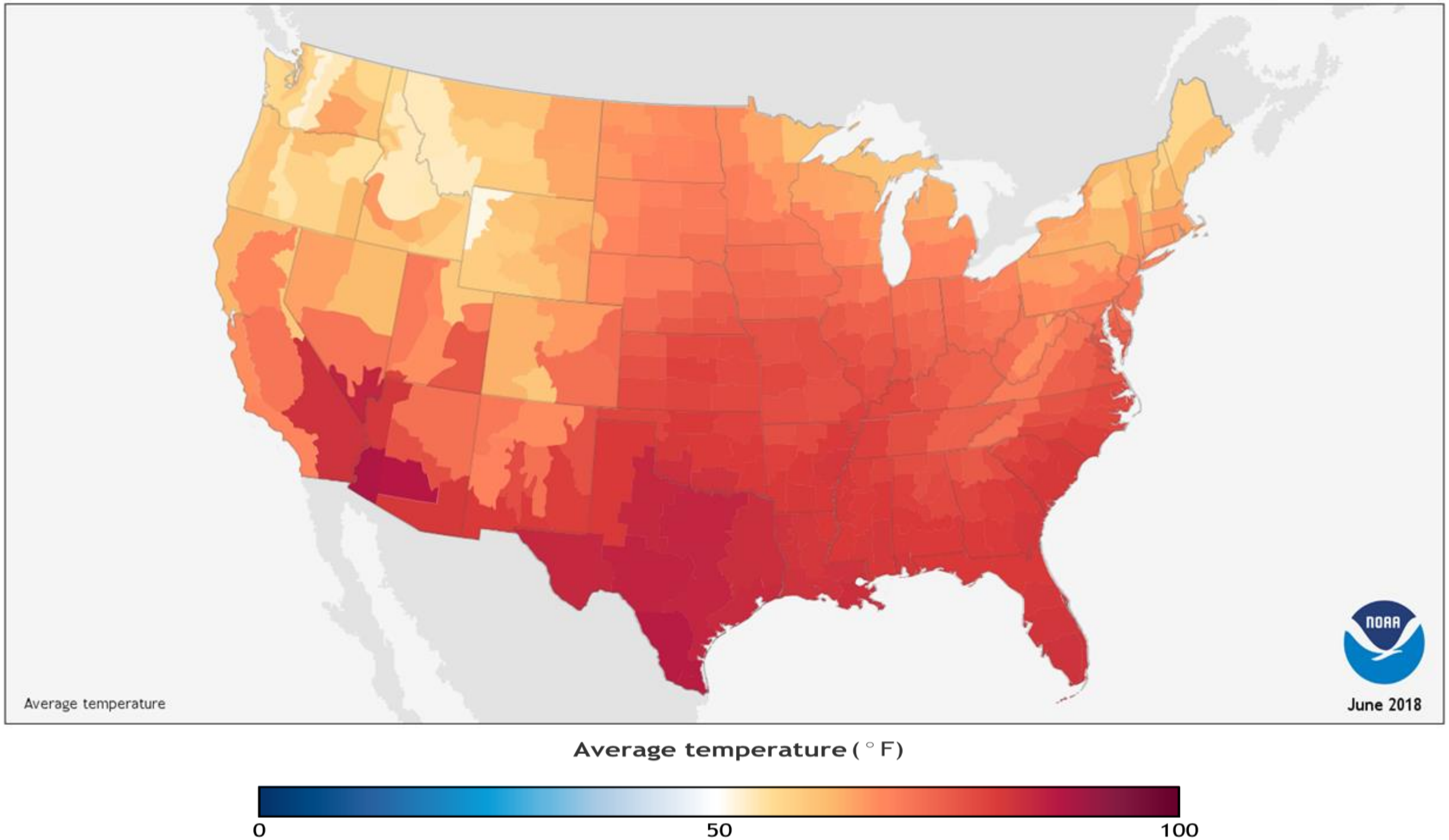
Credit: The Royal Society Publishing

Double Pendulum and Chaos



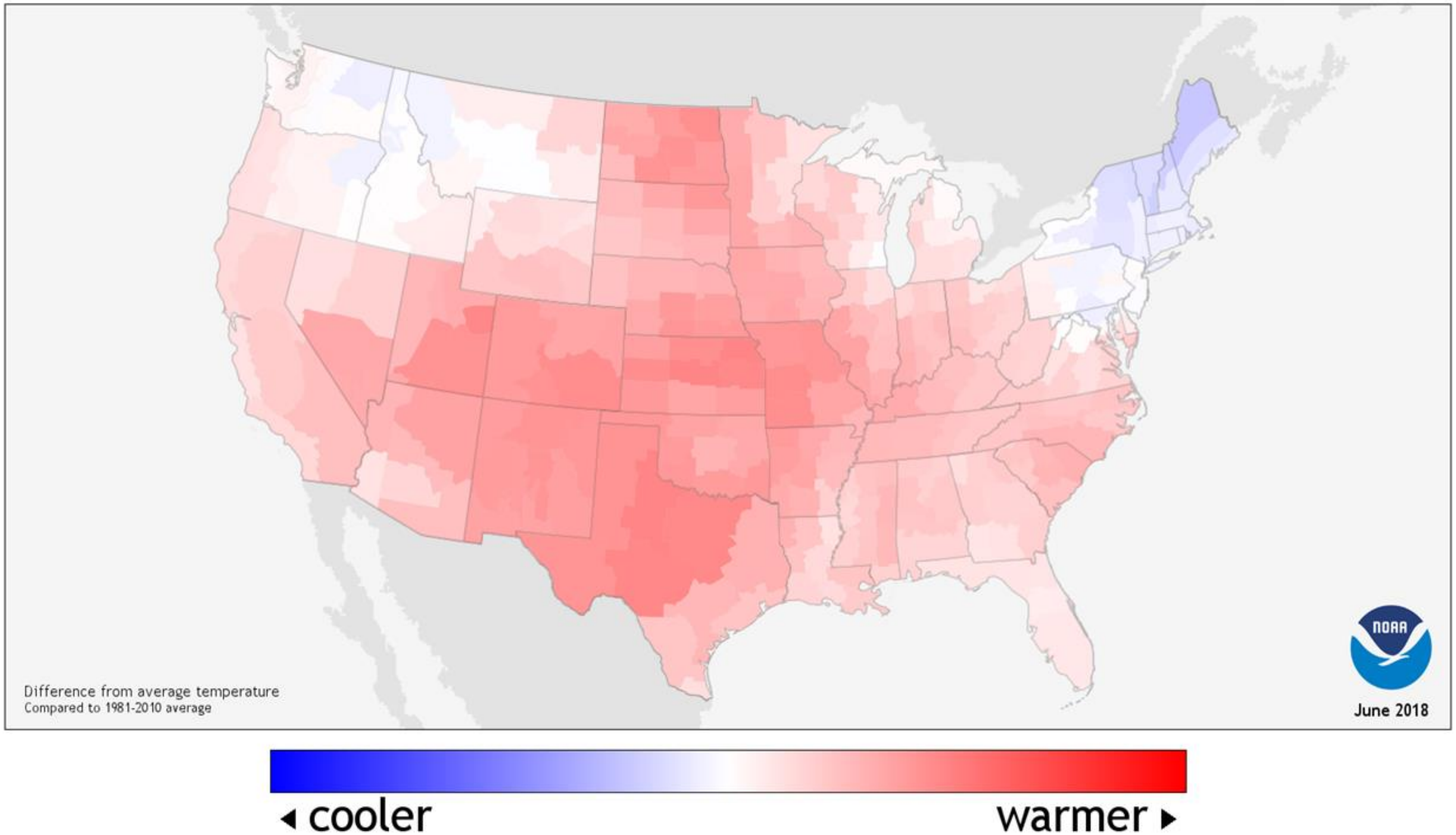
Credit: Wolfram Community

Why do we care about heat events?



June 2018 Average Temperature, Credit: NOAA.gov

Why do we care about heat events?

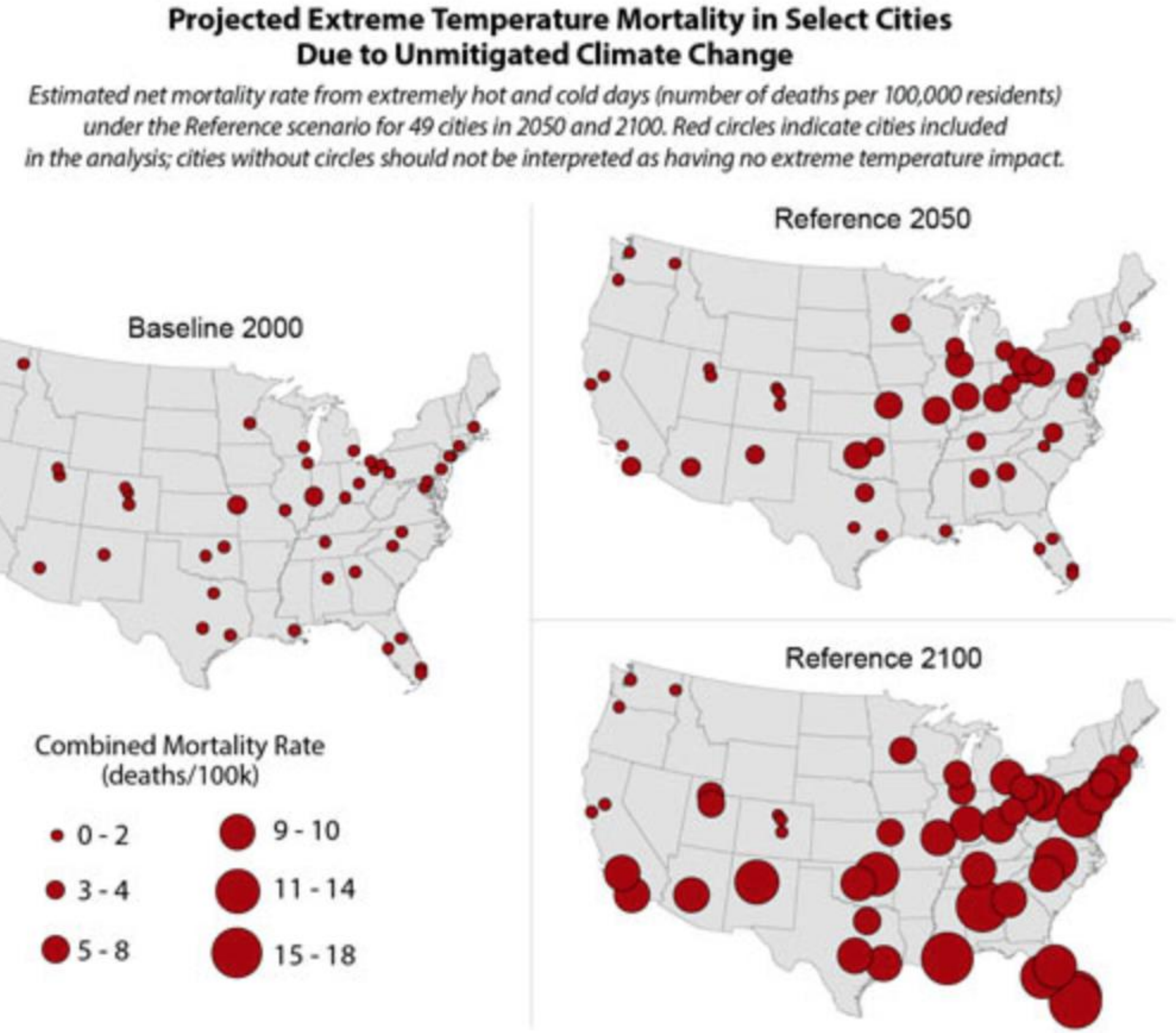


June 2018 Temperature anomalies, Credit: NOAA.gov

Why do we care about heat events?

Credit: IEG Vu

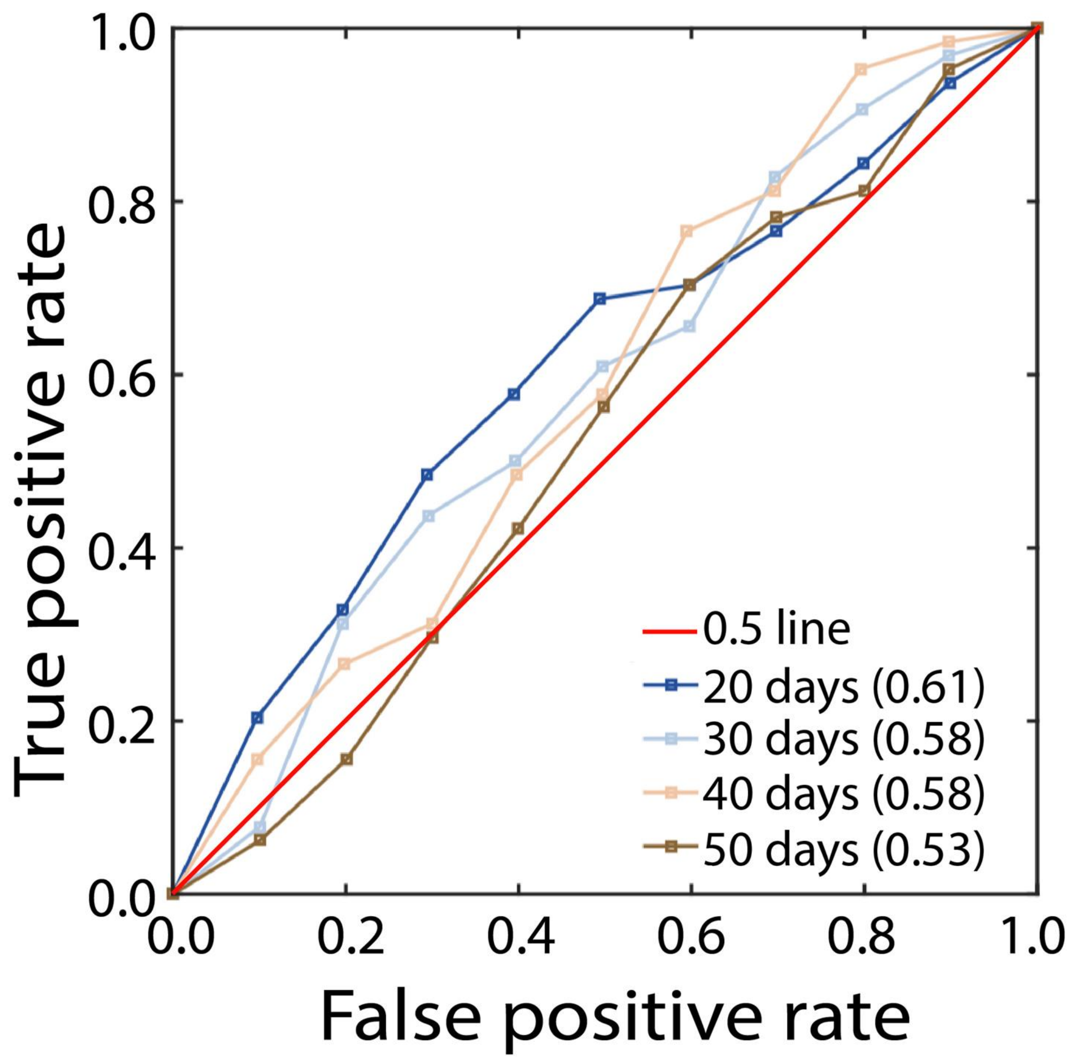
Why do we care about heat events?



Credit: EPA.gov

Background

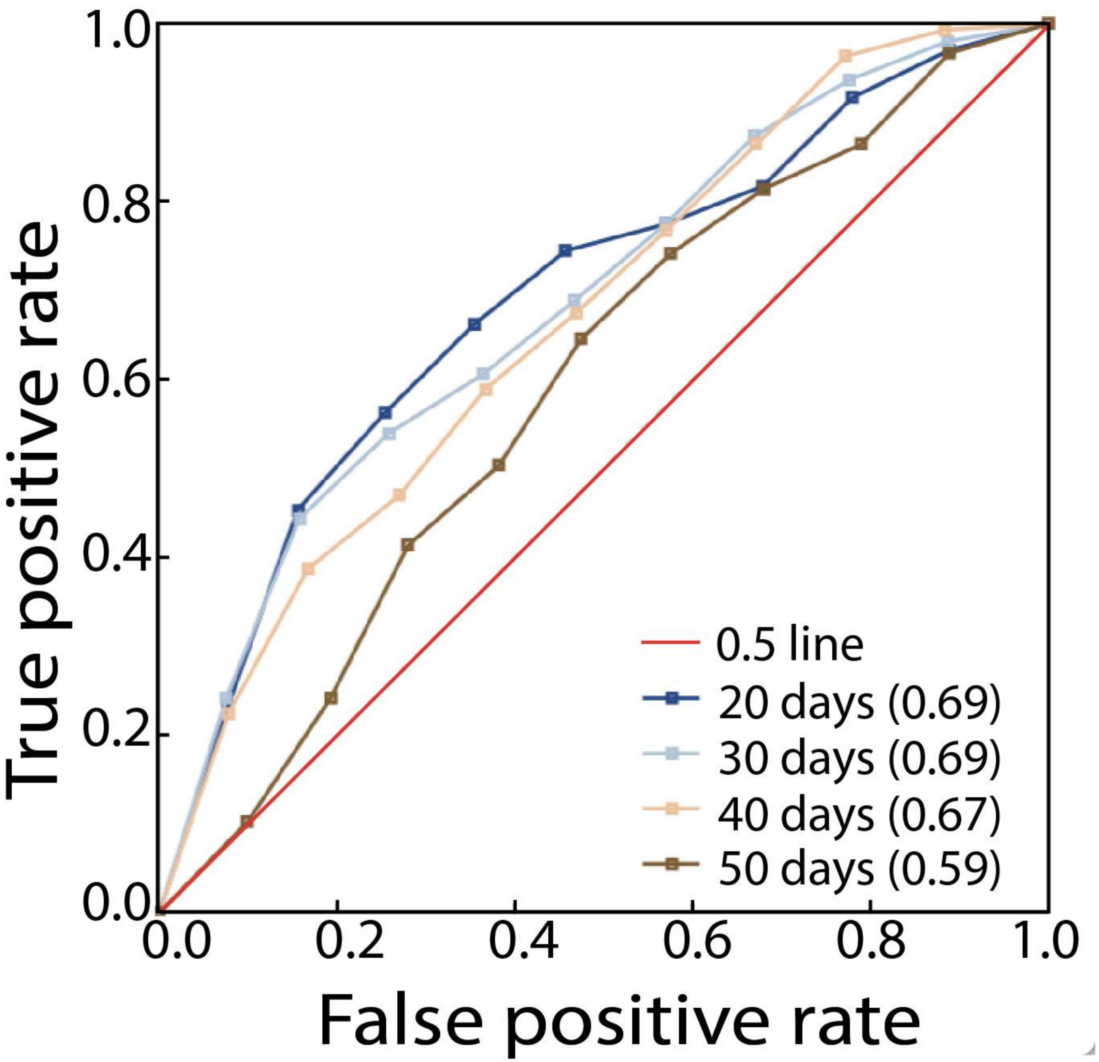
Previous work showed correlation between anomalously warm Sea Surface Temperatures (SST) and anomalously hot days in the Eastern US.



Credit: McKinnon et al, 2015

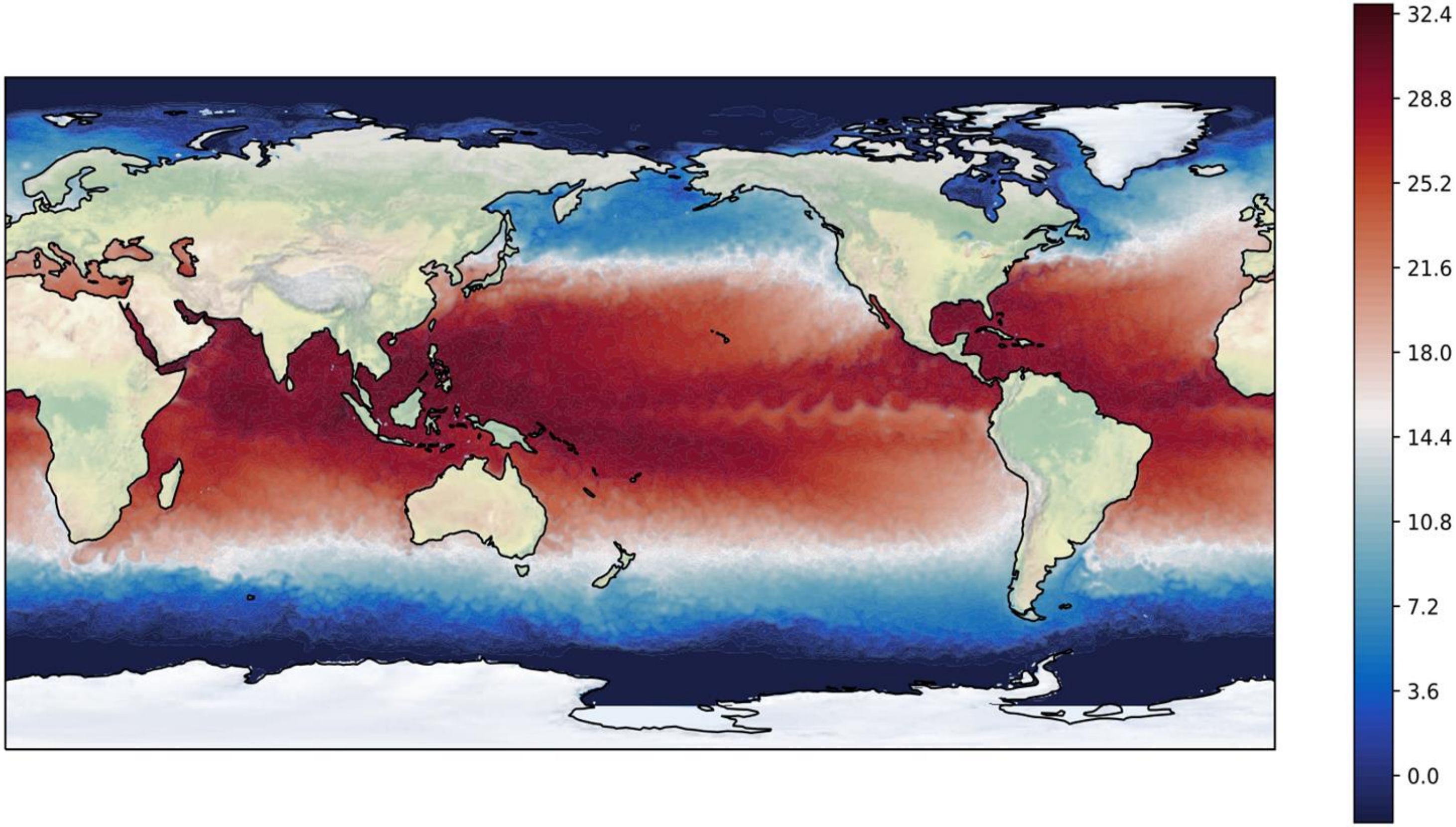
Background

Previous work showed correlation between anomalously warm Sea Surface Temperatures (SST) and anomalously hot days in the Eastern US.



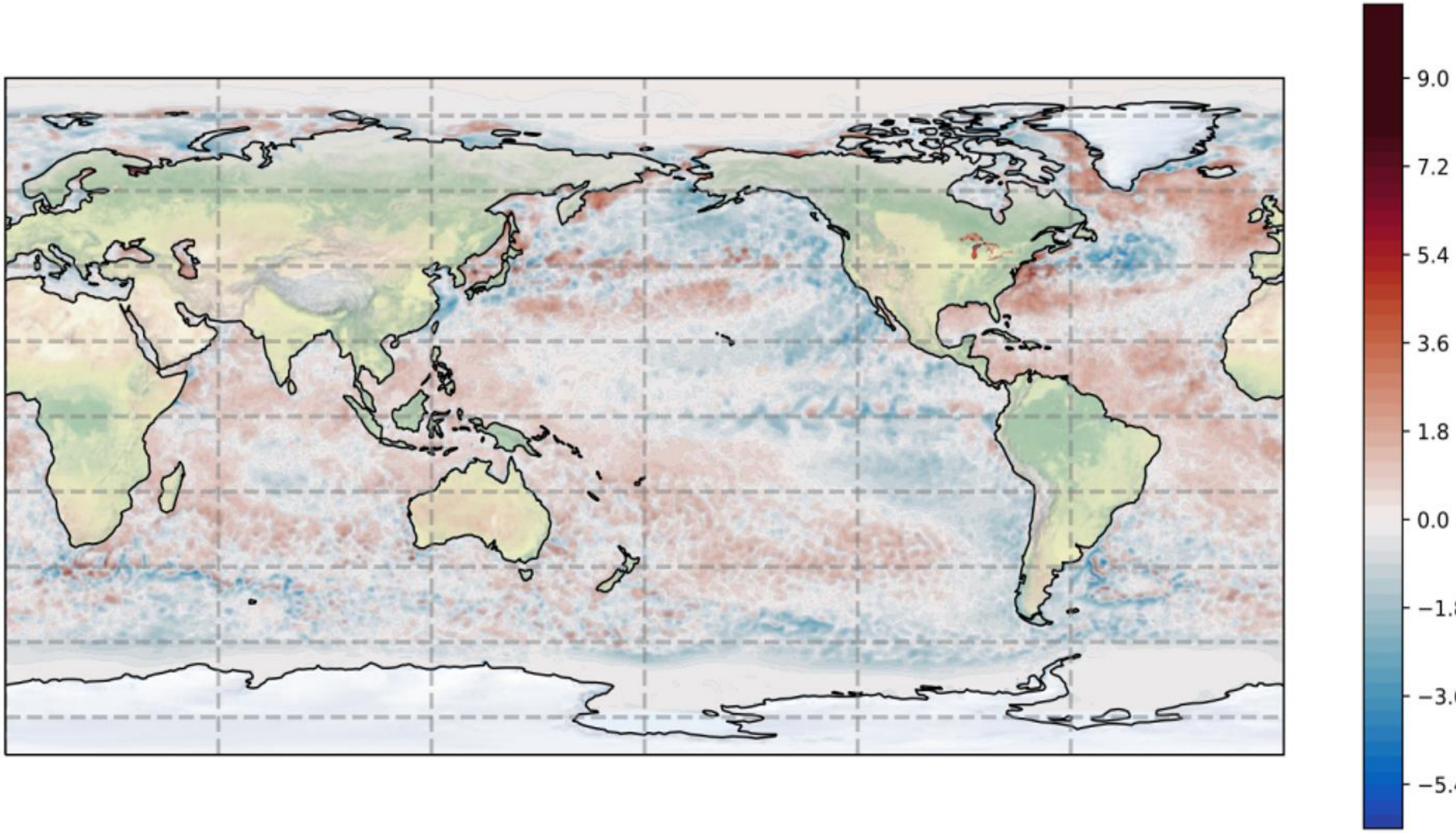
Credit: McKinnon et al, 2015

From Correlation to Statistical Value



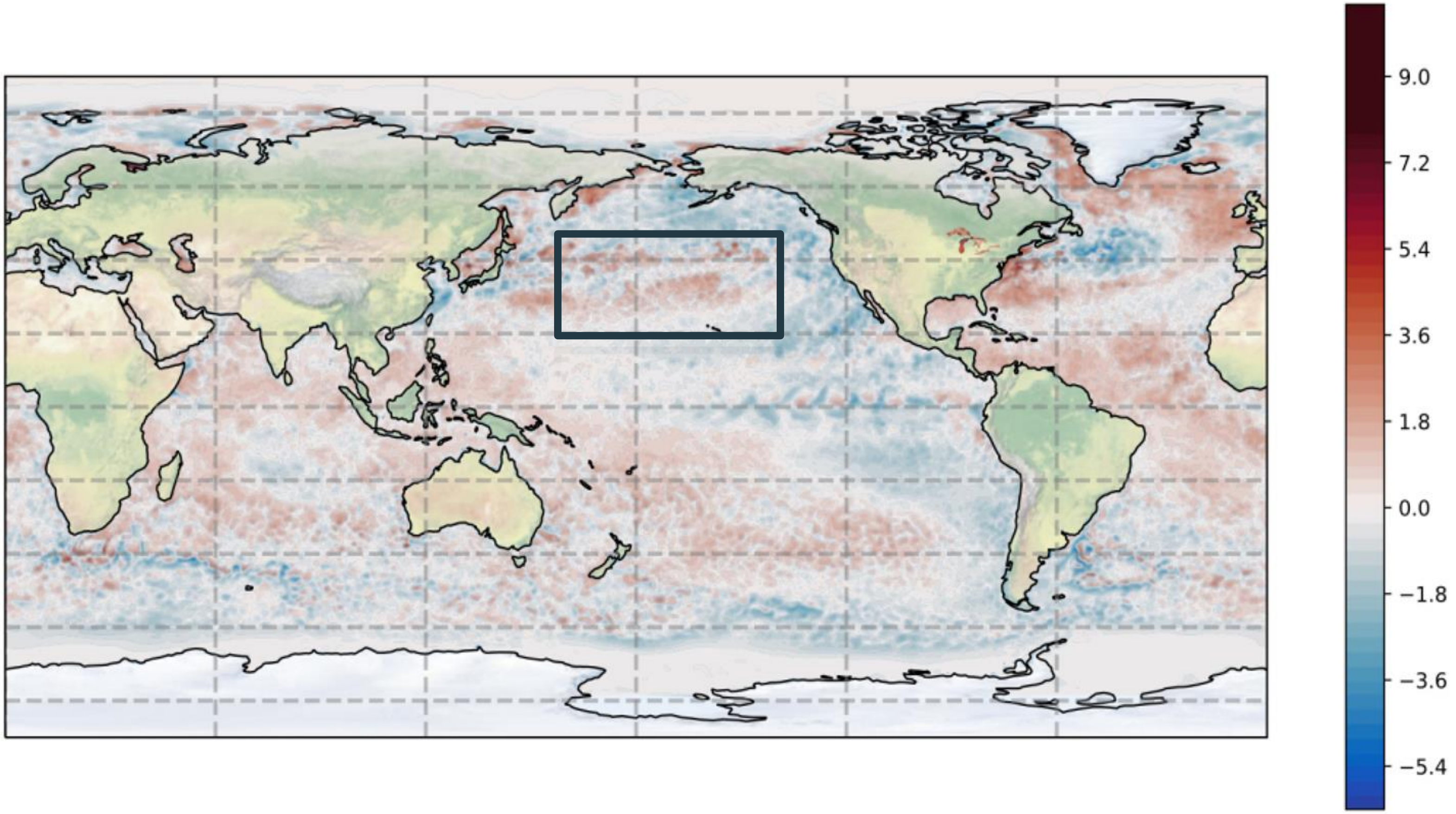
Raw SST data. June 24, 2010

From Correlation to Statistical Value



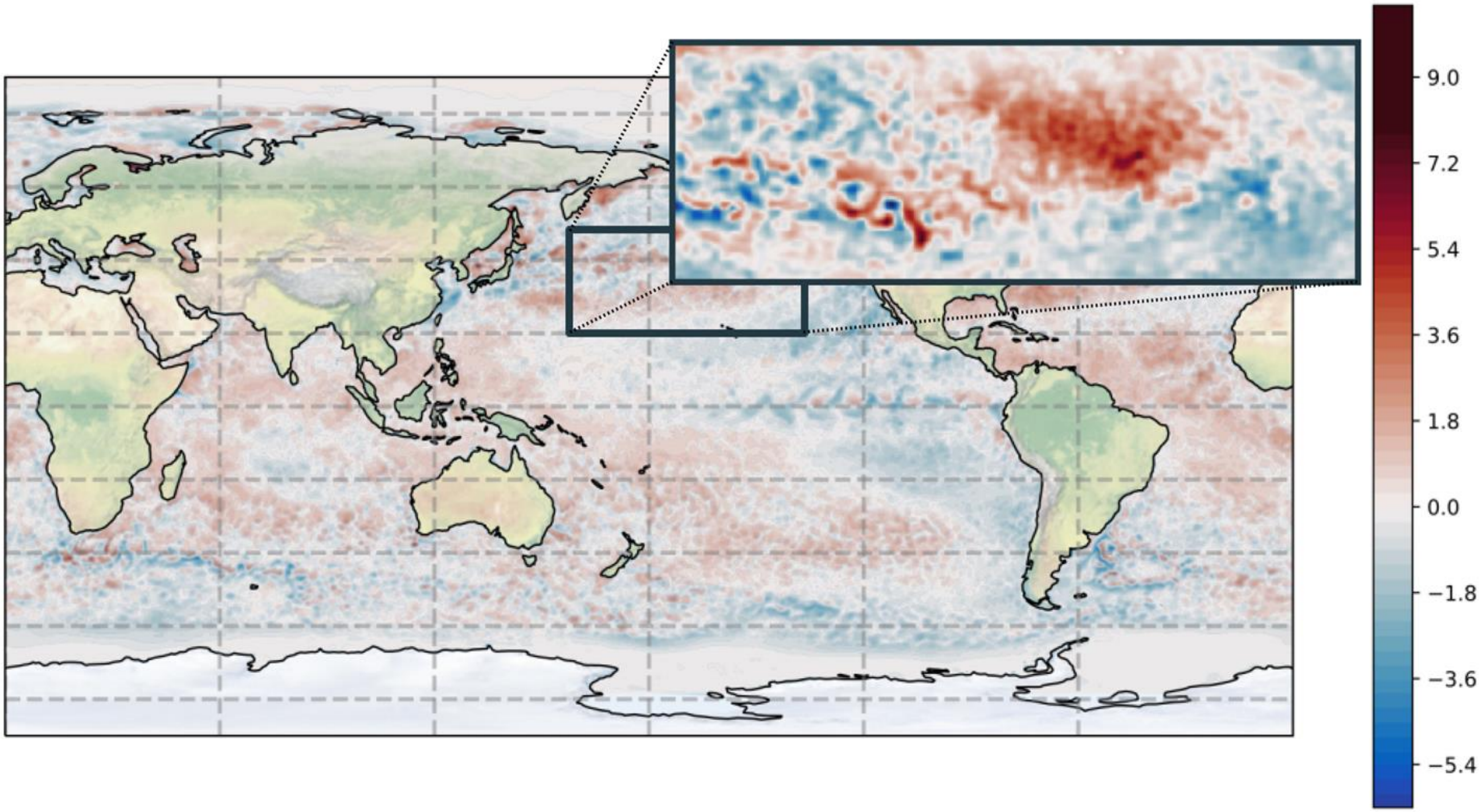
SST anomalies. June 24, 2010

From Correlation to Statistical Value



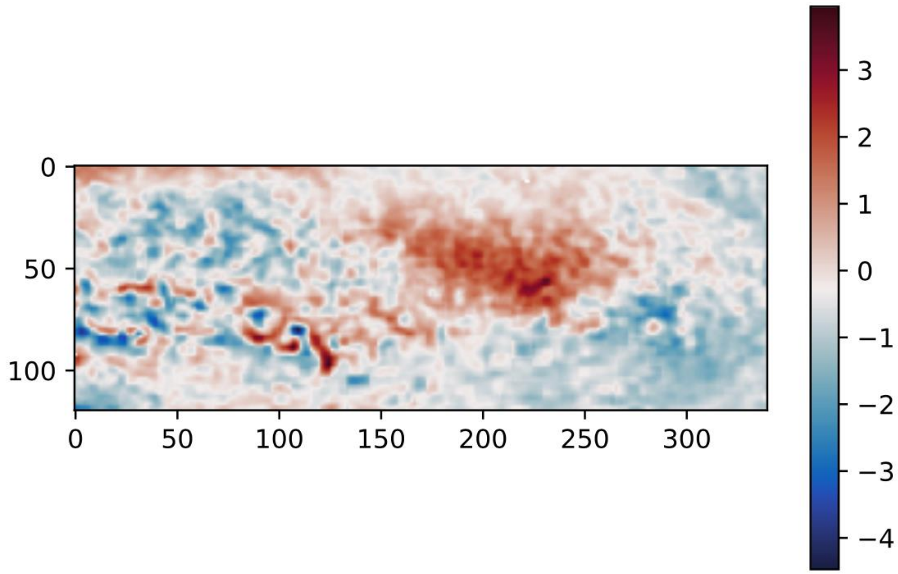
SST anomalies. June 24, 2010

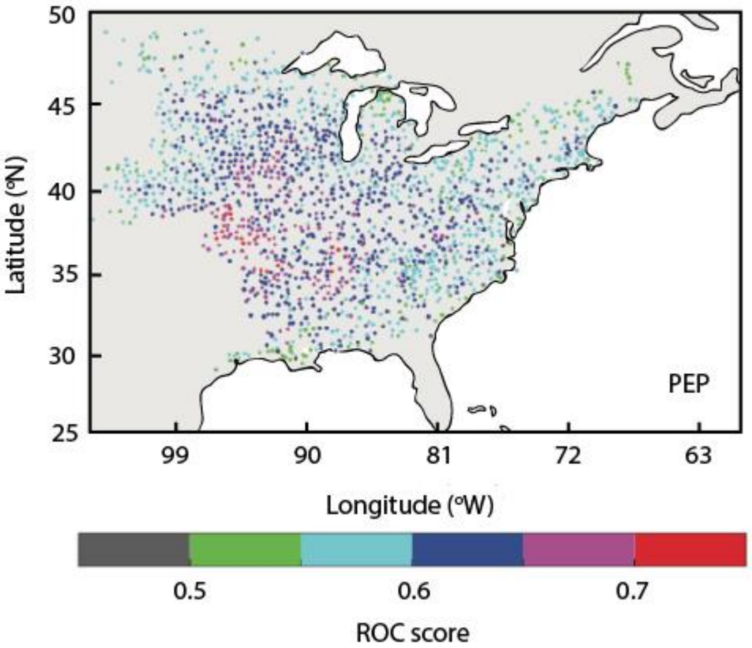
From Correlation to Statistical Value



SST anomalies. June 24, 2010

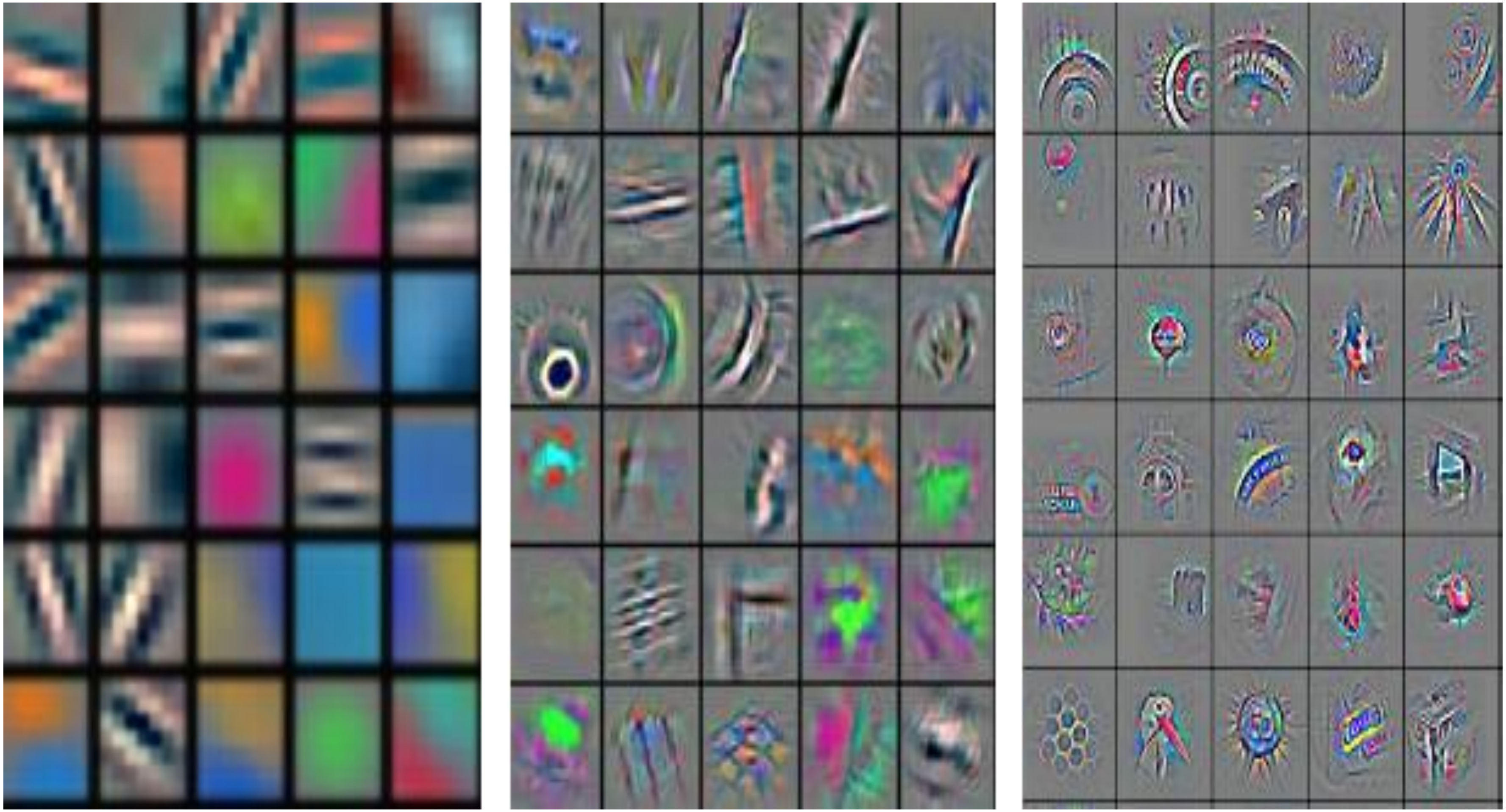
Input and Output





Credit: McKinnon et. al 2015 [edited for clarity]

Why Neural Networks?

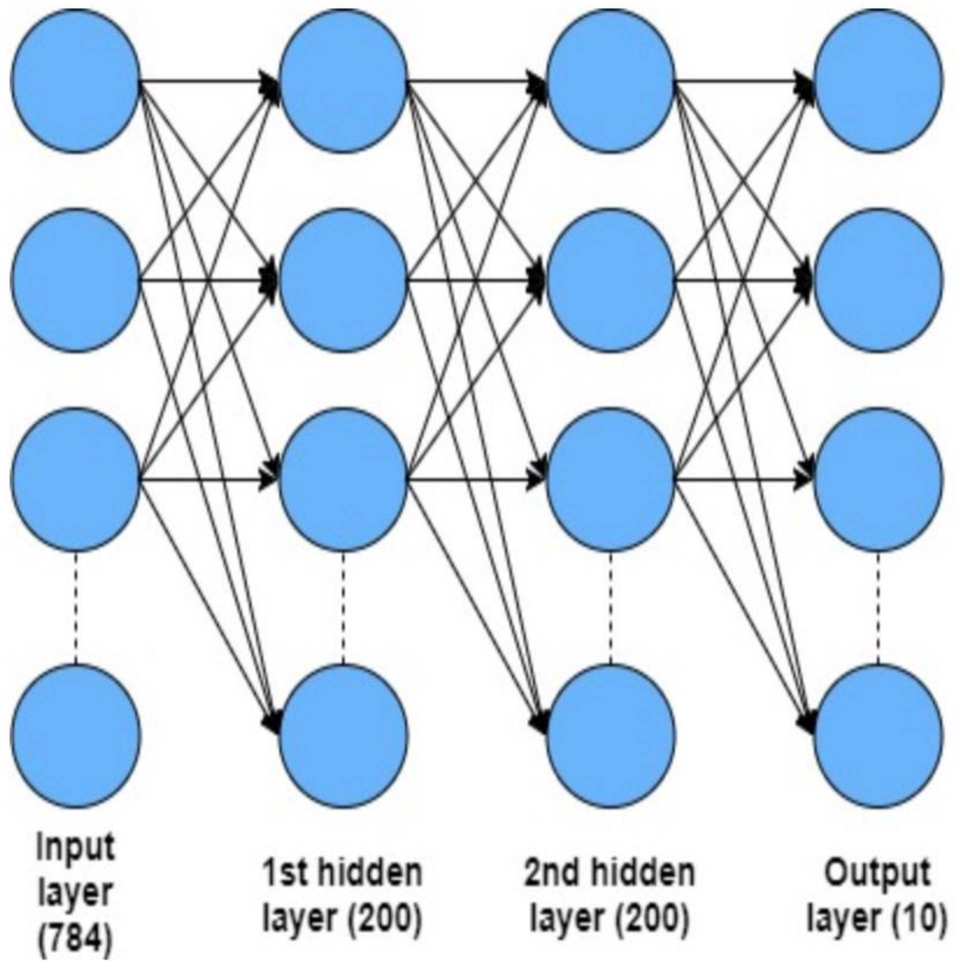


Credit: Fast.ai

Dense Net

- Universal Function Approximators
- Can really (over)learn anything with enough layers and neurons

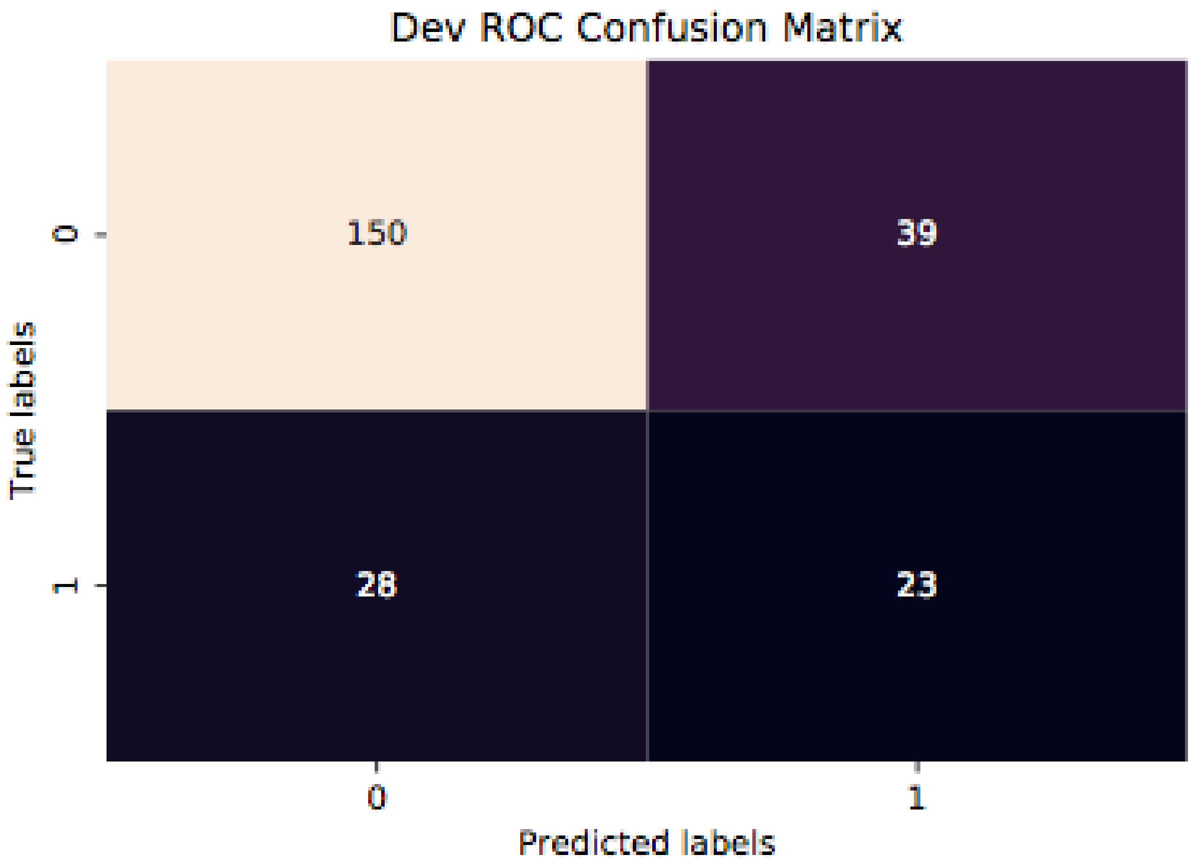
Credit: Wikipedia



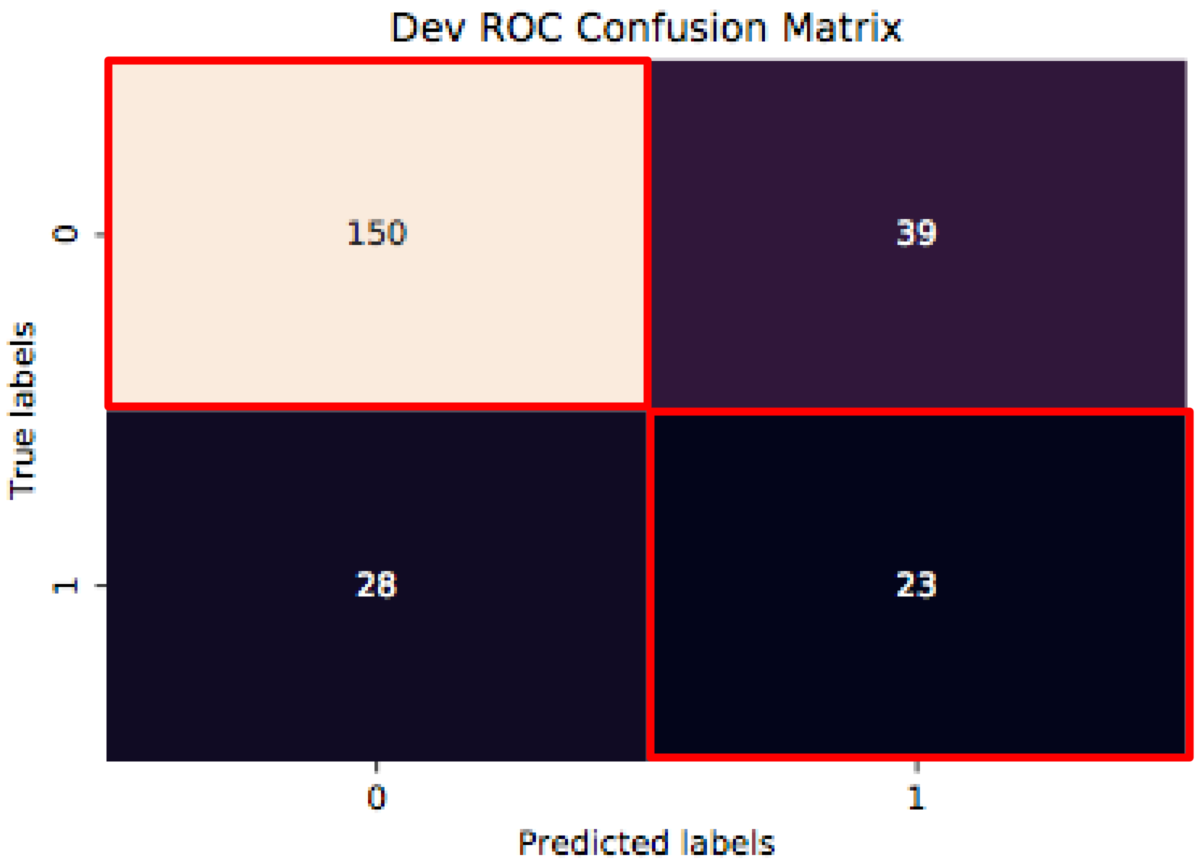
Fully connected neural network example architecture

Credit: Adventures in Machine Learning

Confusion matrix for Best Model



Confusion matrix for Best Model



Acknowledgements

Davide del Vento

Negin Sobhani, Dave Stepaniak, Alessandro Fanfarillo

AJ Lauer, Cecilia Banner, Elliot Foust, Jenna Preston

Molly Winslow

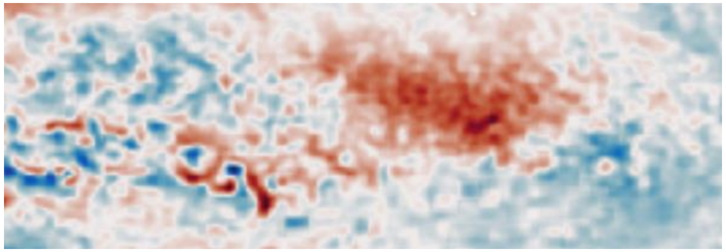
UCAR/NCAR

SIParCS, CISL

NOAA ESRL

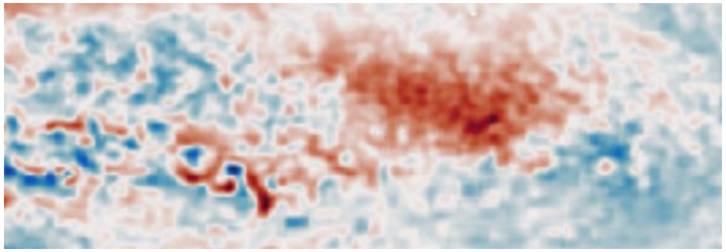
NSF

Methods Overview

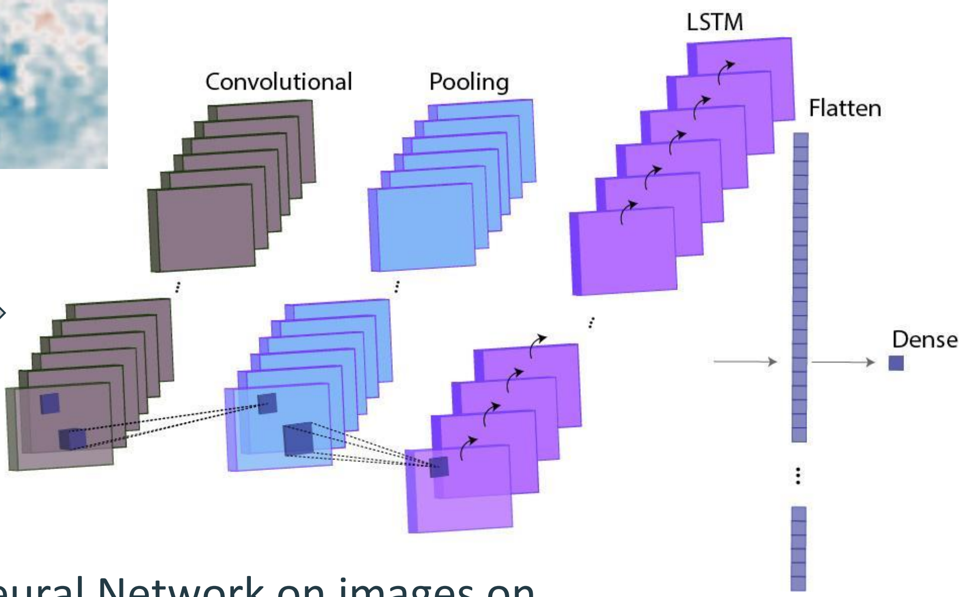


Create images of SST in Pacific

Methods Overview

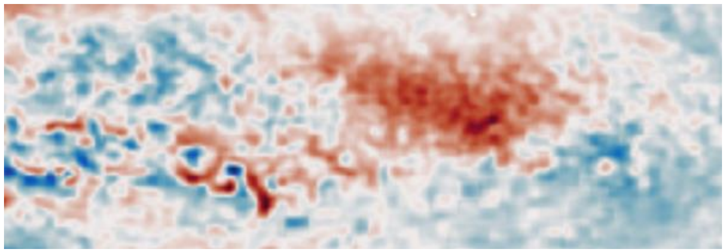


Create images of SST in Pacific

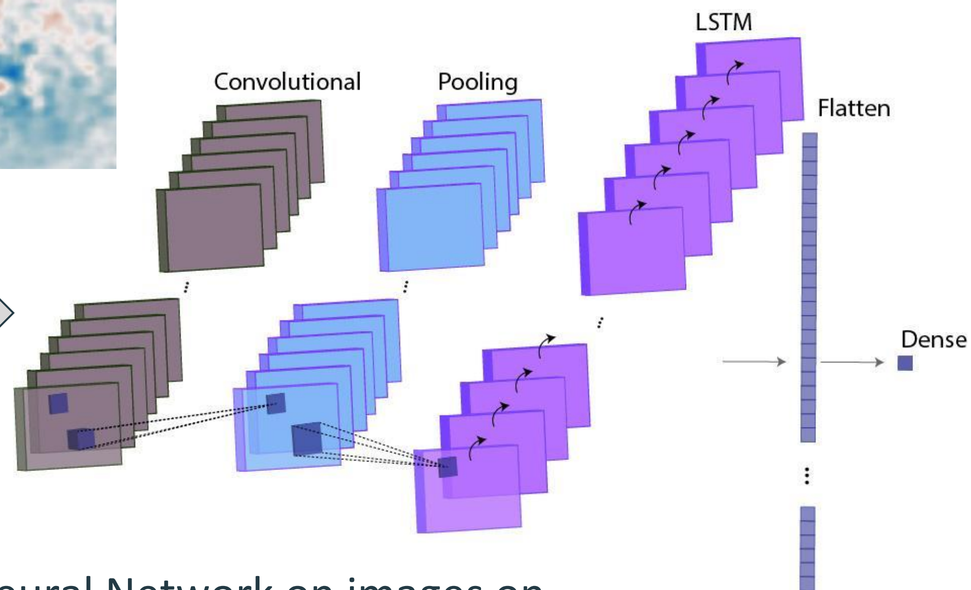


Train Neural Network on images on GPU

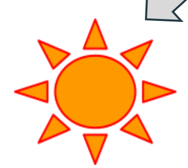
Methods Overview

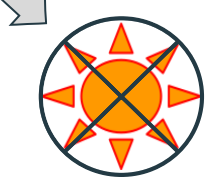


Create images of SST in Pacific



Train Neural Network on images on GPU





Output hot/not hot

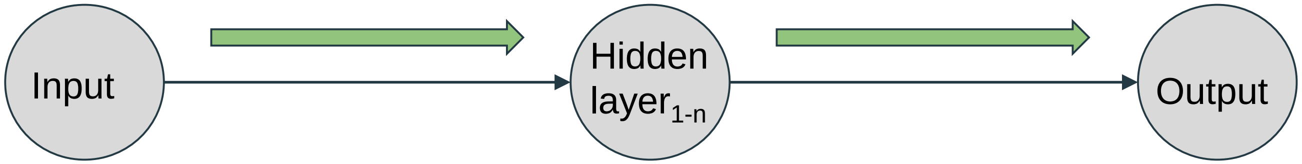
Why Use a GPU?

Due to differences in architecture, GPUs trained the Networks faster with more accuracy than the CPUs.

	Average ROC score	Seconds/Epoch
2.6-GHz Intel XeonE5-2670 CPU	0.44	5.48
NVIDIA K80 GPU	0.55	0.89
Average Increase	11%	615%

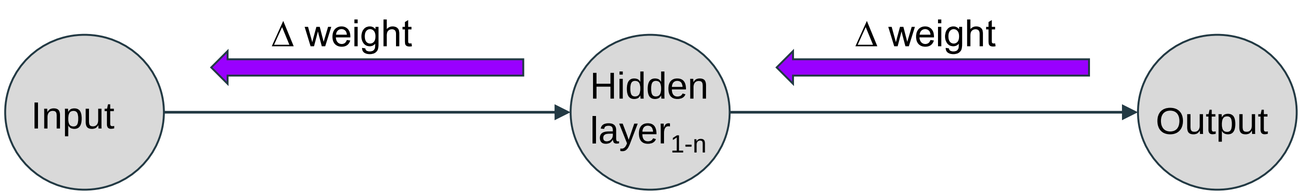
Basic steps in deep learning

- **Forward propagation**
  - Pass input values forward through the network



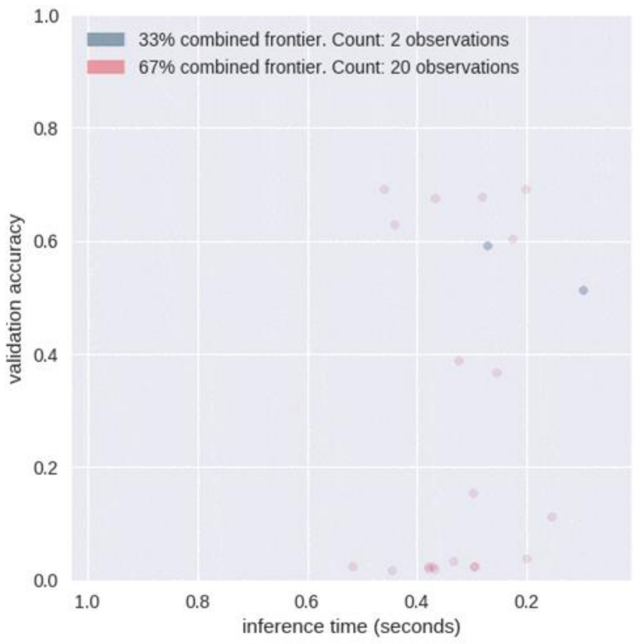
Basic steps in deep learning

- **Forward propagation**
  - Pass input values forward through the network
- **Backward propagation**
  - Adjust weights between neurons
  - minimize loss function



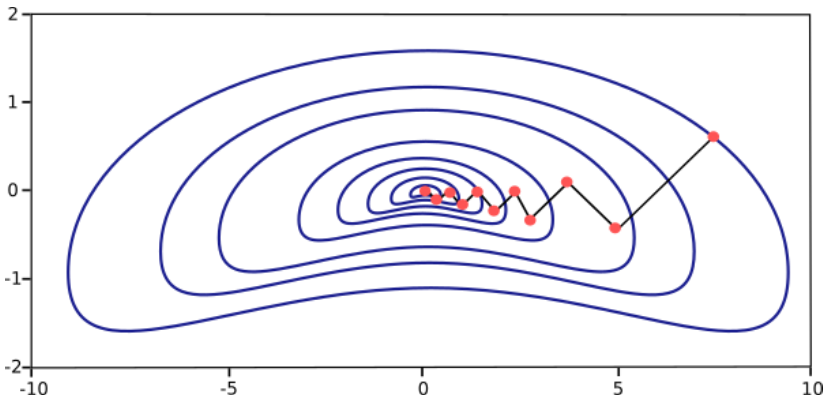
Basic steps in deep learning

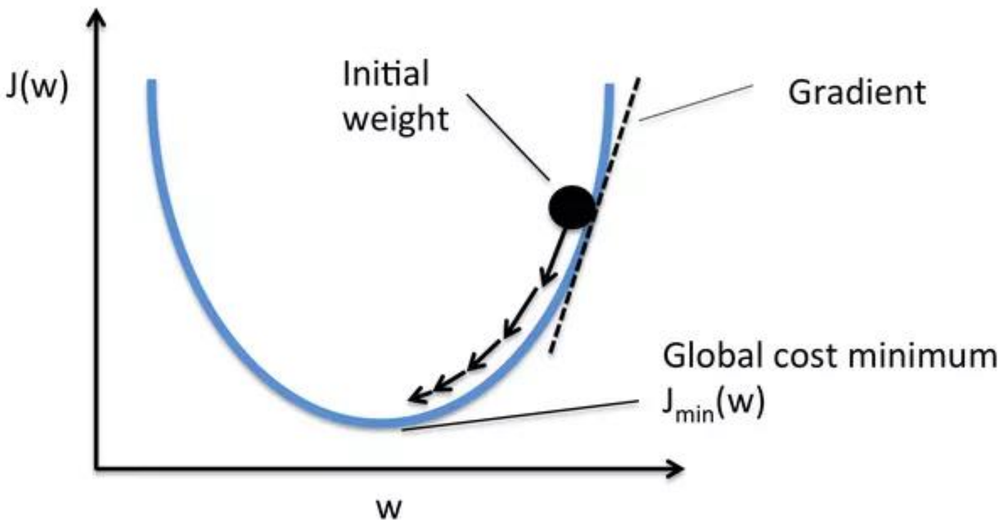
- **Forward propagation**
  - Pass input values forward through the network
- **Backward propagation**
  - Adjust weights between neurons
  - minimize loss function
- **Hyperparameter optimization**
  - Change values such as learning rate and momentum (used in Backward propagation)
  - Can help minimize the loss function



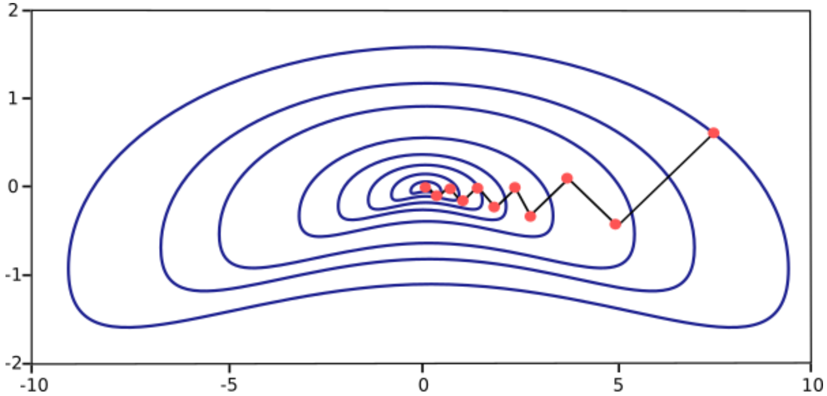
Credit: Nvidia

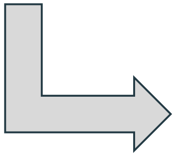
Loss Function

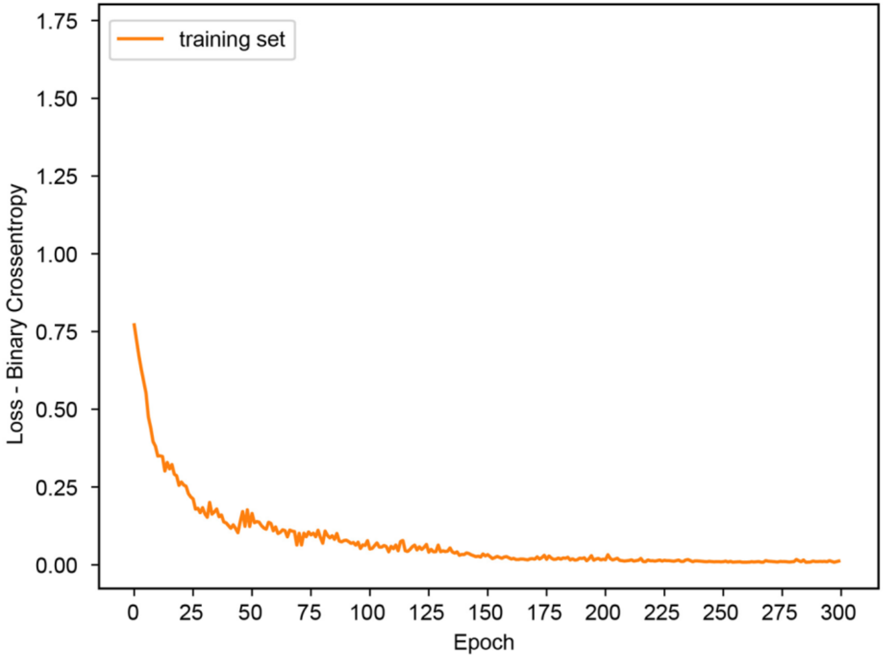




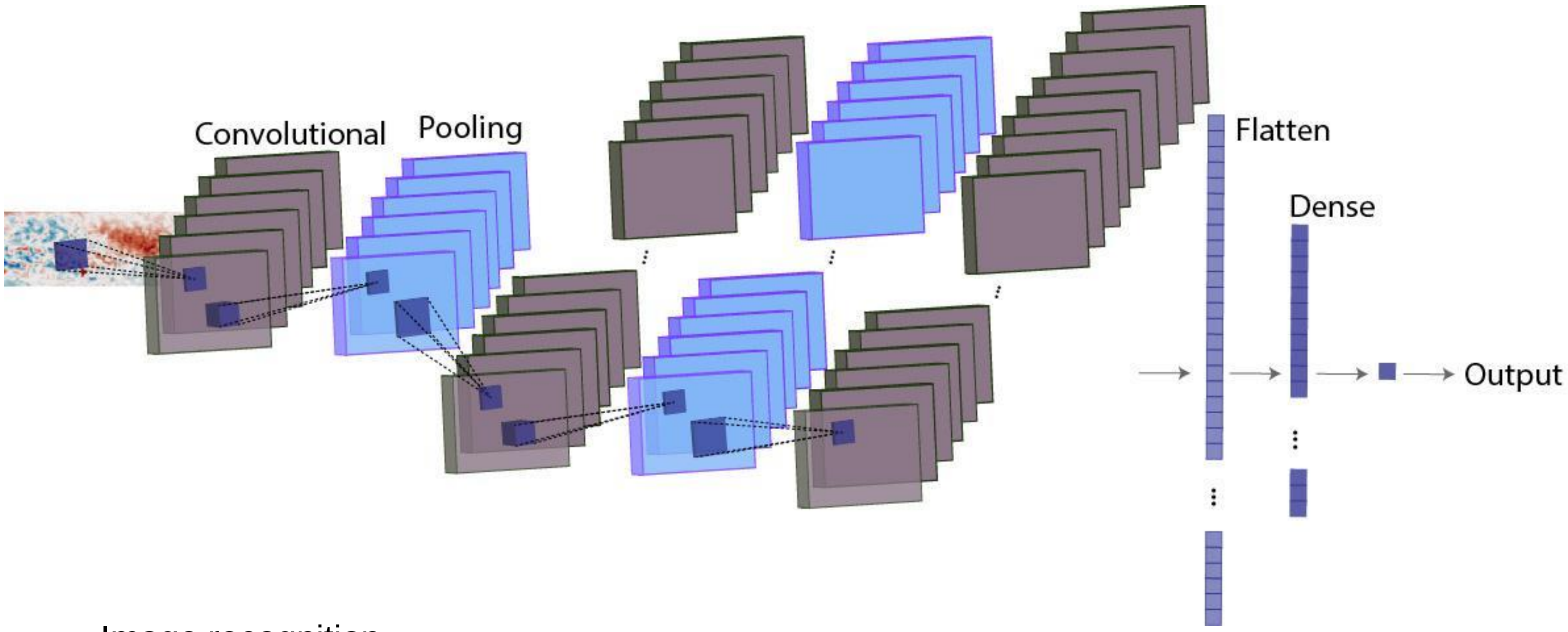
Loss Function







Convolutional network



- Image recognition
- Video Analysis
- Training AI agents to play games
- Facial recgnition

Convolution

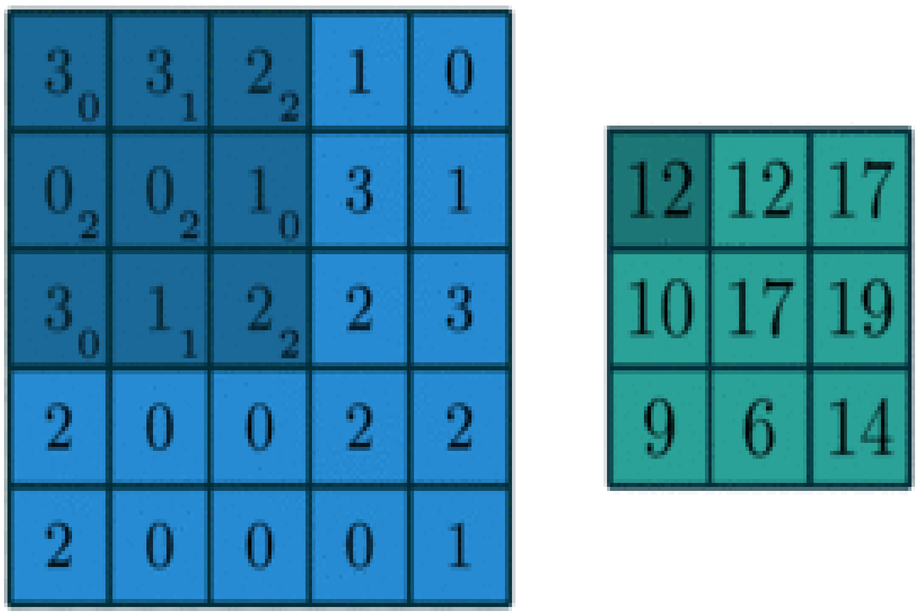
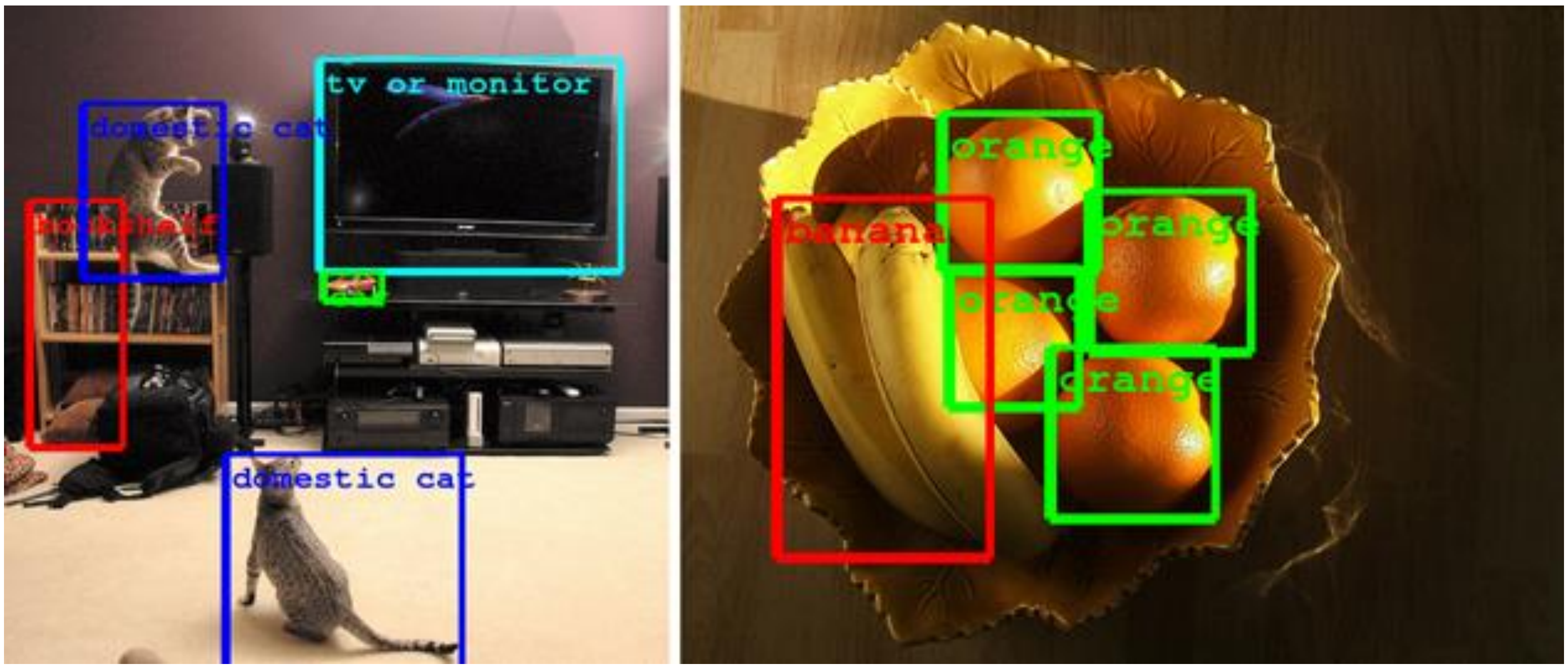
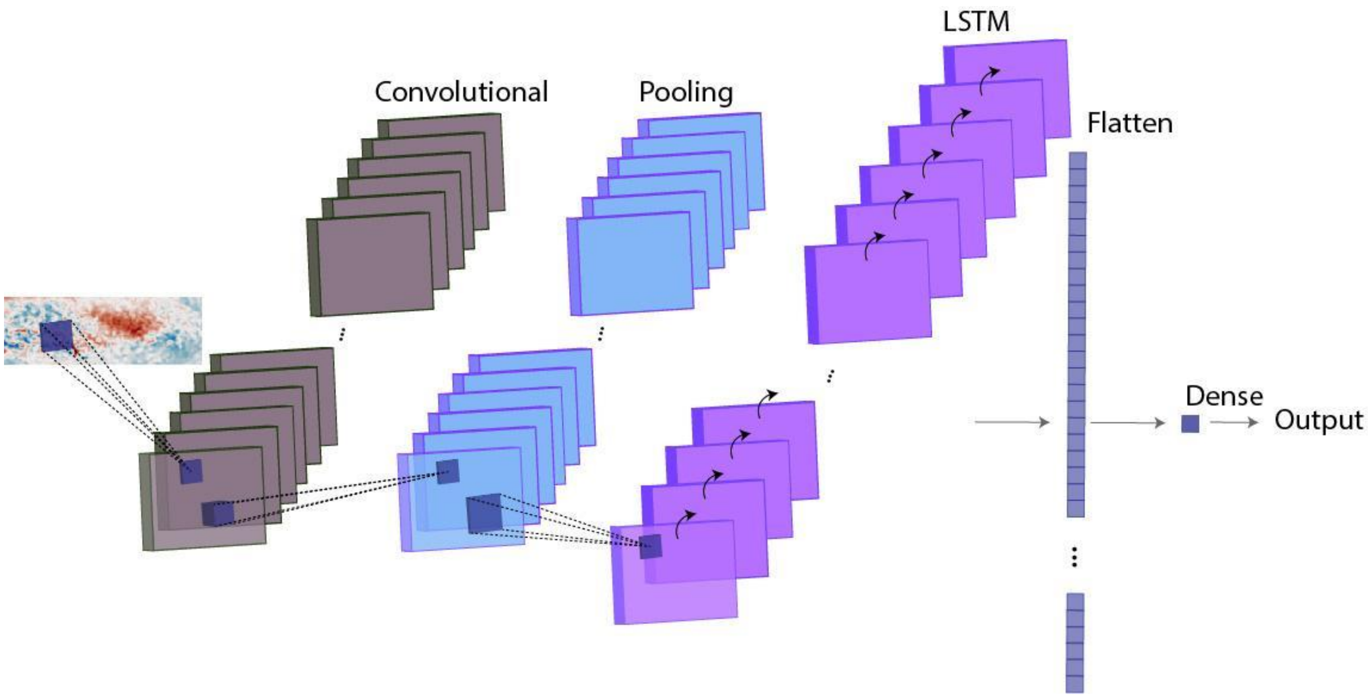


Image Recognition Example



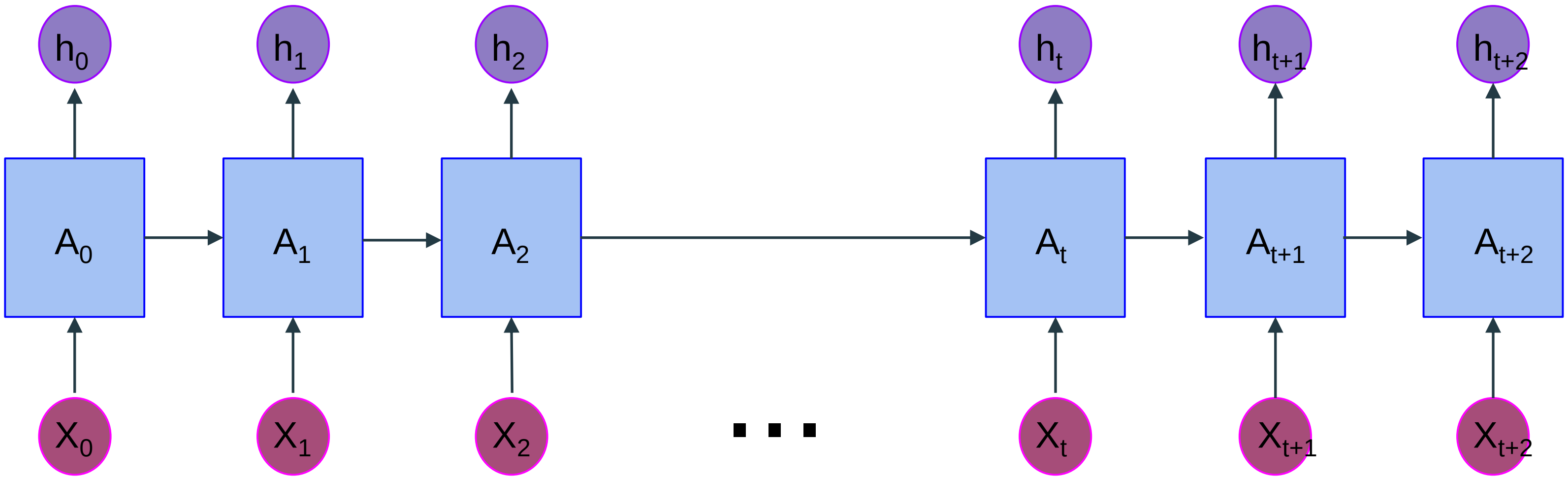
https://www.engadget.com/2014/09/08/google-details-object-recognition-tech/

Long Short Term Memory (LSTM) Network

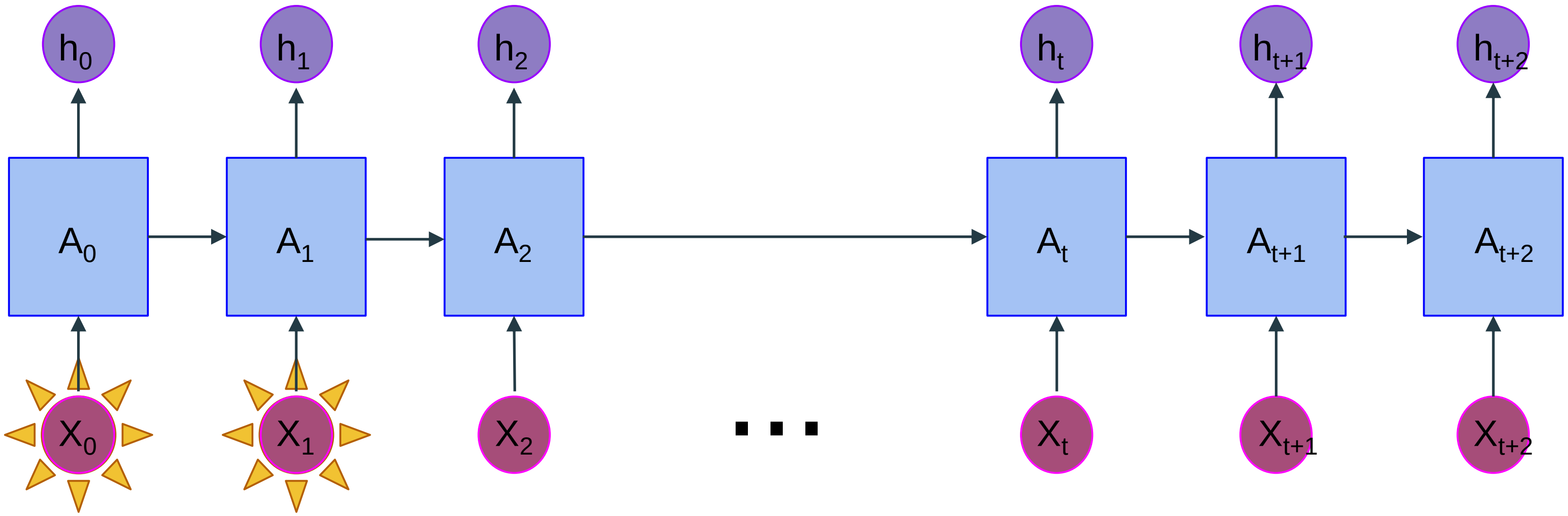


- Time series prediction
- Speech recognition
- Rhythm learning
- Music composition
- Grammar learning
- Handwriting recognition
- Human action recognition

LSTM layer

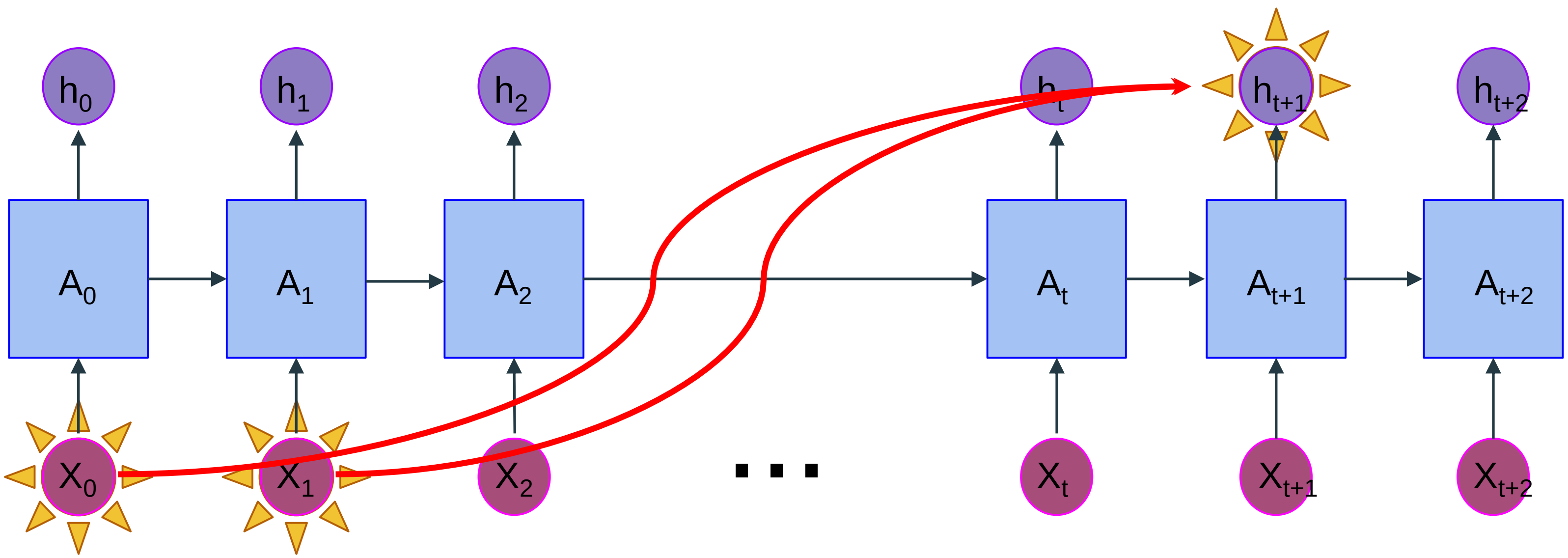


LSTM layer



LSTM layers can manage long-term dependencies.

LSTM layer



LSTM layers can manage long-term dependencies.

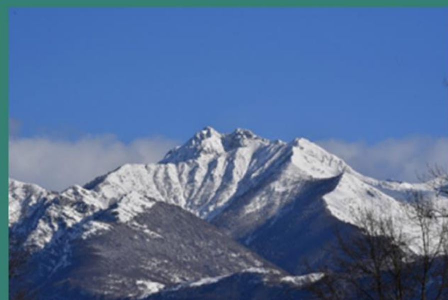
LSTM example: Image Captioning

I think it’s a snow covered mountain.

https://www.captionbot.ai/

LSTM example: Image Captioning

I think it’s a snow covered mountain.

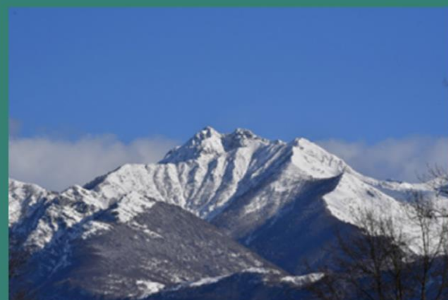


I think it’s a man wearing a hat and sunglasses talking on a cell phone.

https://www.captionbot.ai/

LSTM example: Image Captioning

I think it's a snow covered mountain.

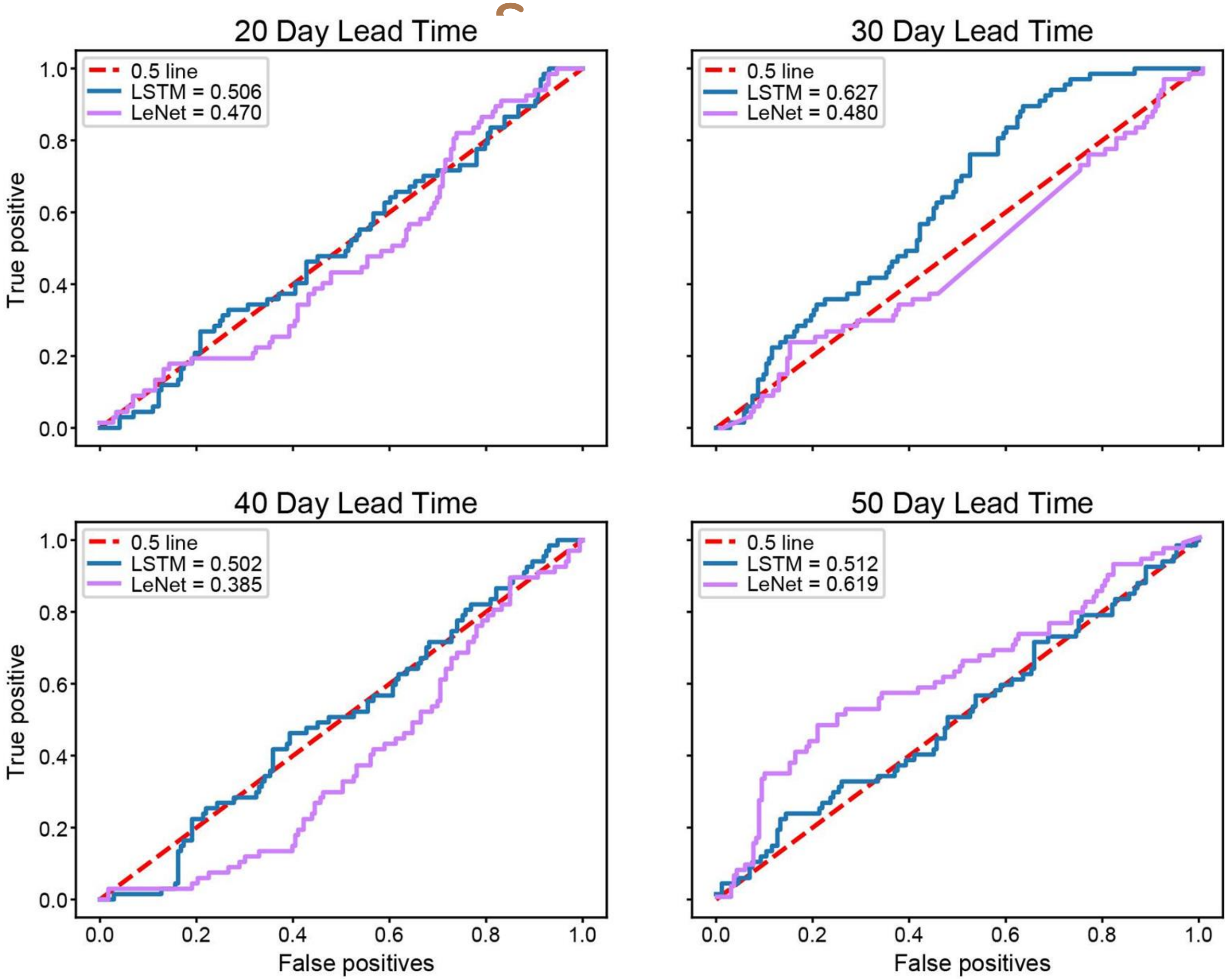


I think it's a dog sitting in front of a fence.

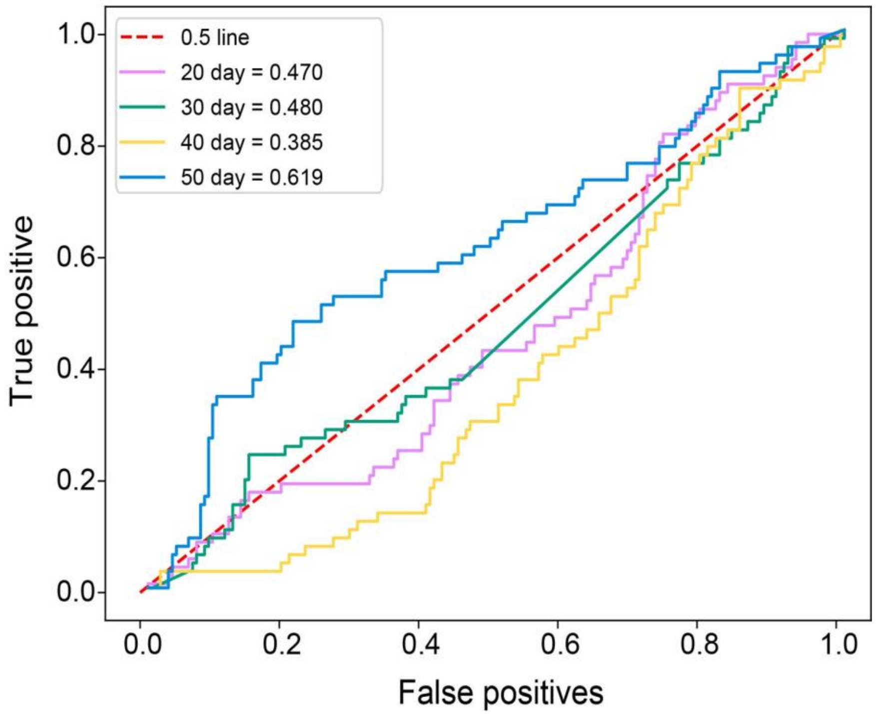
I think it's a man wearing a hat and sunglasses talking on a cell phone.

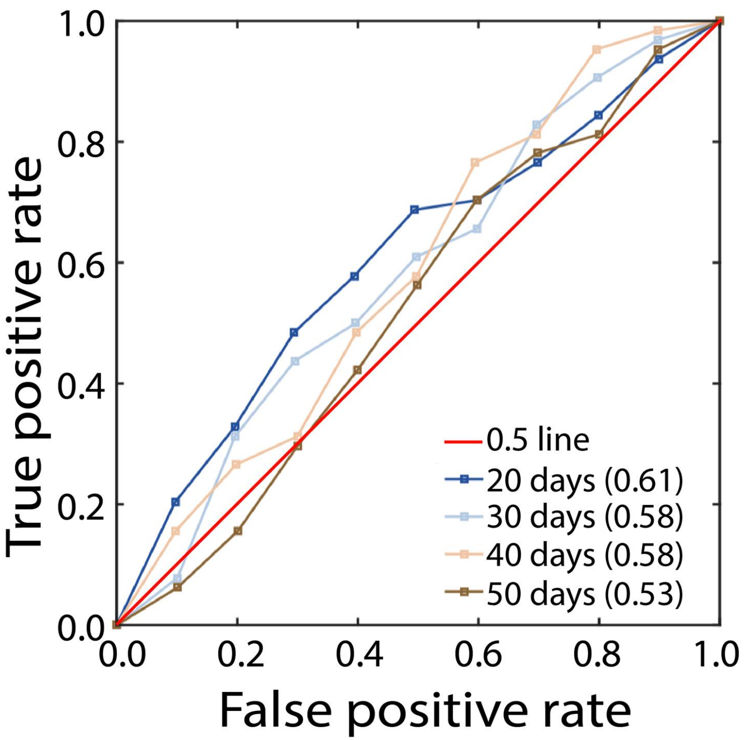
https://www.captionbot.ai/

Results

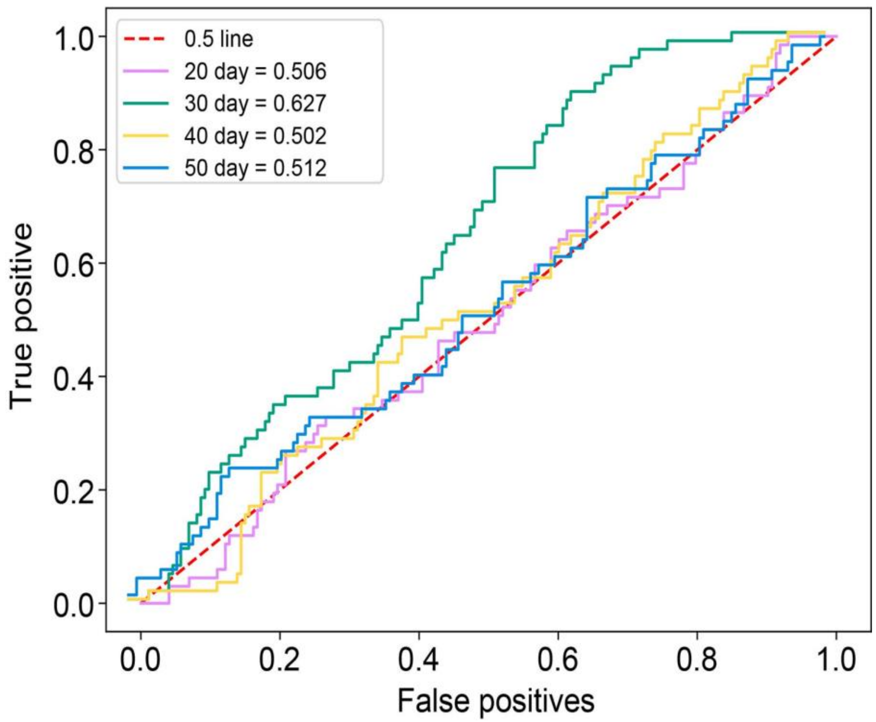

20 Day Lead Time
0.5 line
LSTM = 0.506
LeNet = 0.470
30 Day Lead Time
0.5 line
LSTM = 0.627
LeNet = 0.480
40 Day Lead Time
0.5 line
LSTM = 0.502
LeNet = 0.385
50 Day Lead Time
0.5 line
LSTM = 0.512
LeNet = 0.619
True positive
False positives
0.0
0.2
0.4
0.6
0.8
1.0

Convolutional Net ROC

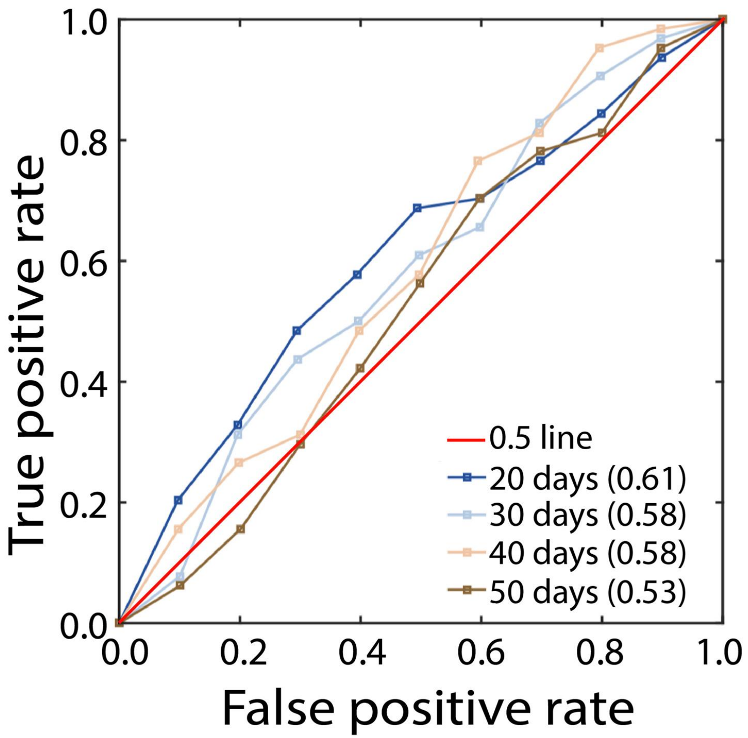




LSTM ROC



0.5 line
20 day = 0.506
30 day = 0.627
40 day = 0.502
50 day = 0.512
True positive
False positives
0.0
0.2
0.4
0.6
0.8
1.0



0.5 line
20 days (0.61)
30 days (0.58)
40 days (0.58)
50 days (0.53)
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0

Conclusions

- GPUs were faster at training both Networks than CPUs
- The LSTM network performed better overall than the LeNet network
- The LSTM network predicts better than random chance but is only significantly better for the 30 day lead time

Future Work

- Recreate McKinnon's week long prediction
- Finish optimizing networks for better ROC scores
- Additional architectures

Acknowledgements

Davide del Vento

Negin Sobhani, Dave Stepaniak, Alessandro Fanfarillo

AJ Lauer, Cecilia Banner, Elliot Foust, Jenna Preston

UCAR/NCAR

SIParCS, CISL

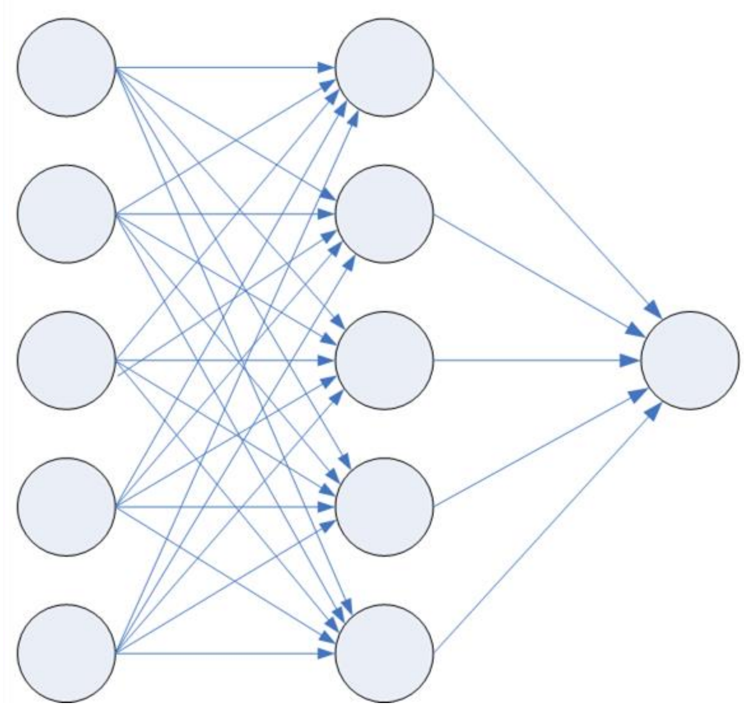
NOAA ESRL

NSF

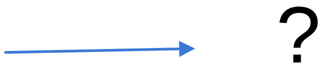
Papers Cited

- McKinnon et. al 2015. Long-lead predictions of eastern United States hot days from Pacific sea surface temperatures
- Gao Huang et al. “Snapshot Ensembles: Train 1, get M for free”. In: *CoRR* abs/1704.00109 (2017). arXiv: 1704.00109. URL: http://arxiv.org/abs/1704.00109.
- Wojciech et al, 2015. Interactive Systems for Designing Machine Elements and Assemblies.
- Han et al, 2017. Pre-Trained AlexNet Architecture with Pyramid Pooling and Supervision for High Spatial Resolution Remote Sensing Image Scene Classification

Questions?



Credit: IBM.com



Data Formatting

- NOAA ESRL High Resolution SST data
- Used Unidata NetCDF module in Python
- Makes .nc file to a MFDataset

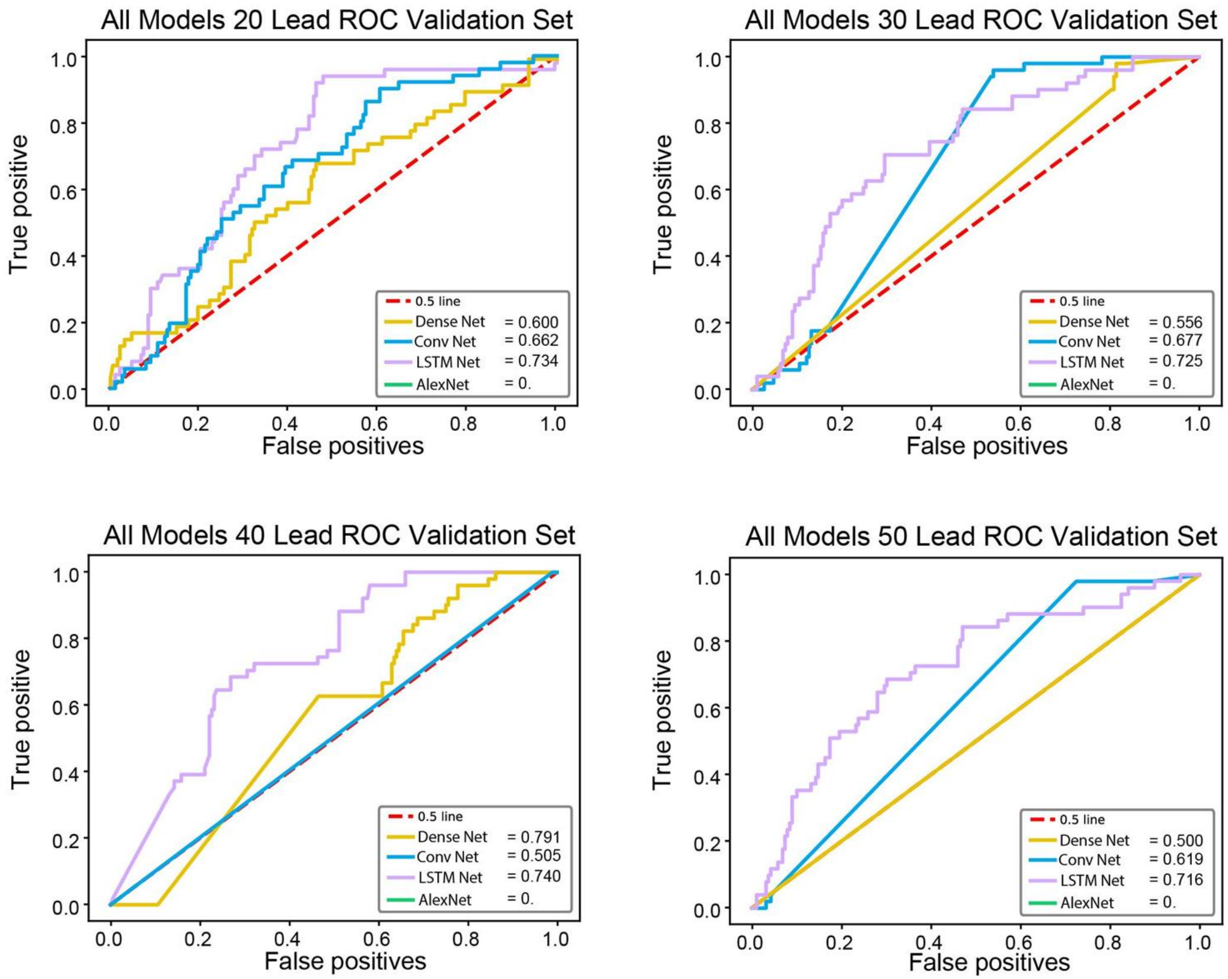
Pooling

1	3	2	9
7	4	1	5
8	5	2	3
4	2	1	4

7	9
8	

Quick explanation

Development set



Optimized Hyperparameters

net	LeNet			
lead time	20	30	40	50
optimizer	SGD	Adam	Adam	SGD
class weight	1	1	3	1
learning rate	0.01	0.01	0.01	0.01
epochs	156	300	178	300
batch size	110	128	164	128
ROC	0.682	0.642	0.688	0.681
model choice	view	view	view	view
net	LSTM			
lead time	20	30	40	50
optimizer	SGD	Adam	Adam	SGD
class weight	3	3	6	1
learning rate	0.005	0.002	0.002	0.01
epochs	300	300	300	300
batch size	89	189	189	45
ROC	0.795	0.77	0.7	0.661