Fighting Fire with AI

Using Artificial Intelligence to Improve Modelling and Decision Making in Wildfire Management

BIRS Talk – November 7, 2017

Mark Crowley

Assistant Professor

Electrical and Computer Engineering

University of Waterloo

Outline

- My Interests
- Artificial Intelligence, Machine Learning, Big Data and all that...
  - Decision Making / Reinforcement Learning
  - Deep Learning
  - Deep RL
    - Why not try both?
- Learning Spatial Dynamics for forest wildfire
  - Method I : Deep RL
  - Method II : LRCNs
- Future Challenges and Opportunities

About Me

- The long-term goal of the my research is to augment human decision making in complex environments in a dependable and transparent way.
- Where this complexity arises from
  - spatial structure
  - streaming data
  - or from the presence of anomalous events.

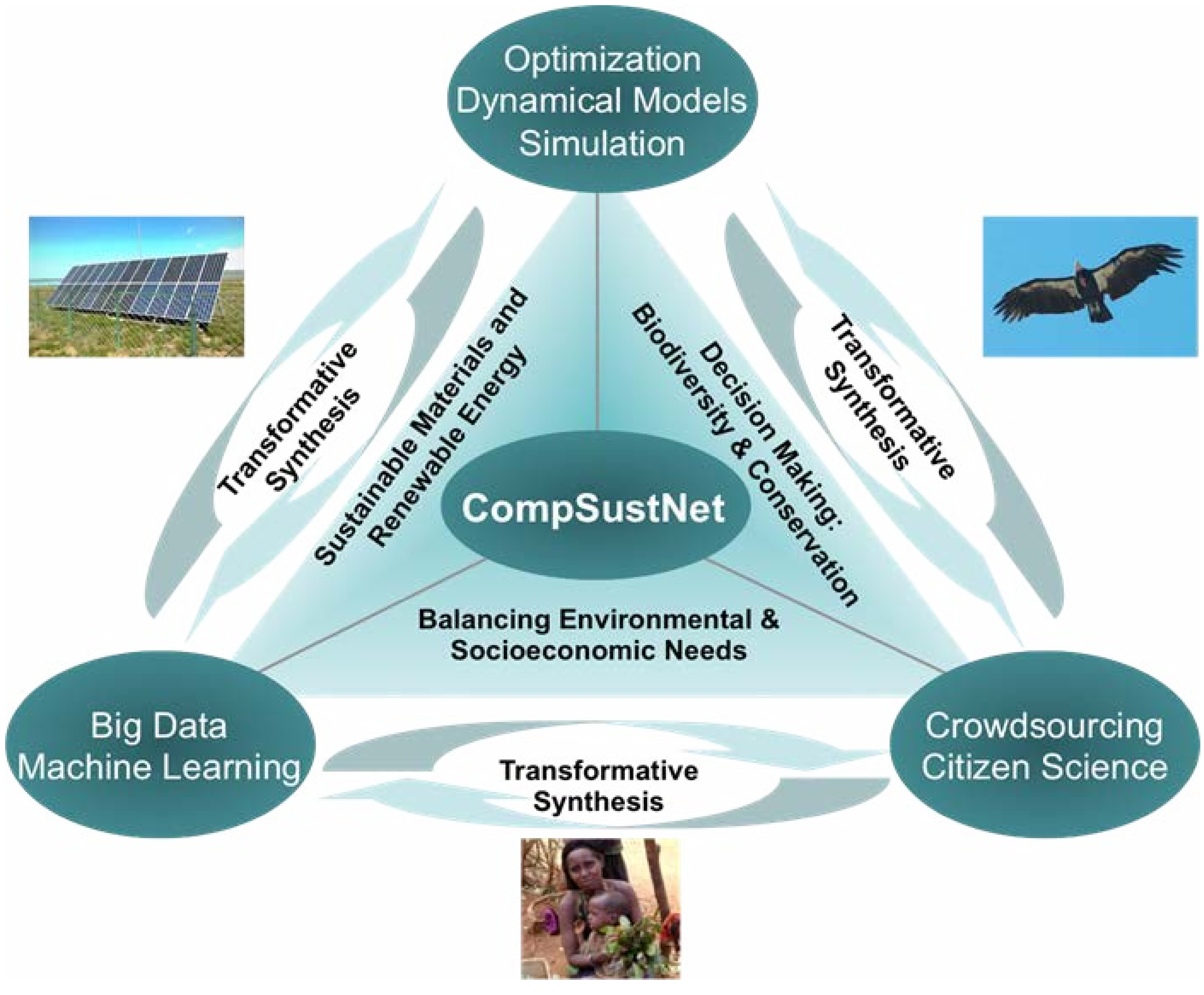
- MSc, PhD in CS from UBC
  - Bayesian network inference
  - Cyclical Causal Models
  - Reinforcement Learning for Sustainable Forest Harvest Management **(David Poole, John Nelson)**
- Postdoc Oregon State University
  - Forest Fire Planning – Treat/Let Burn (**Claire Montgomery**)
  - PAC MDP Planning for Control of Invasive Species (H**eidi Jo Albers**)
  - Computational Sustainability (**Tom Dietterich**) (http://www.compsust.net)
- Waterloo
  - UW ECE ML Lab : https://uwaterloo.ca/scholar/mcrowley/lab
  - ECE Department
  - Pattern Analysis and Machine Intelligence (PAMI)
  - Waterloo Institute for Complexity and Innovation (WICI)

Waterloo Institute for Complexity and Innovation

WATERLOO INSTITUTE
for COMPLEXITY & INNOVATION

- Complex behaviour arises from
  - the interplay in d**ensely interconnected systems**,
  - between multiplicative **causation** and
  - positive and negative **feedbacks**."
  - *i.e."a whole that is more than the sum of its parts"*
- WICI members works on:
  - Coupled human-environment systems
  - Spatial land use change
  - Modelling and planning of urban environment change
  - Social Network Interactions
  - Collaborative Global Governance
  - Dynamics of human impacts on ecosystems
  - Disease spread and vaccine intervention impacts
  - Complex interaction of socio-economics on climate change

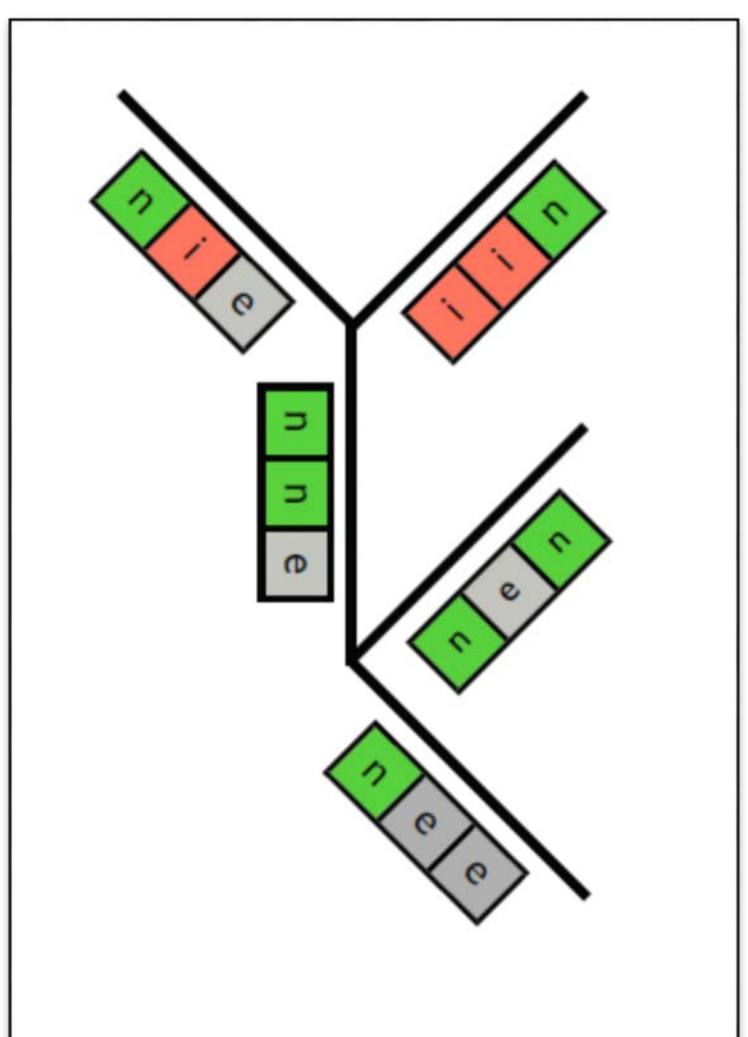
Computational Sustainability



Find out more: http://www.compsust.net/

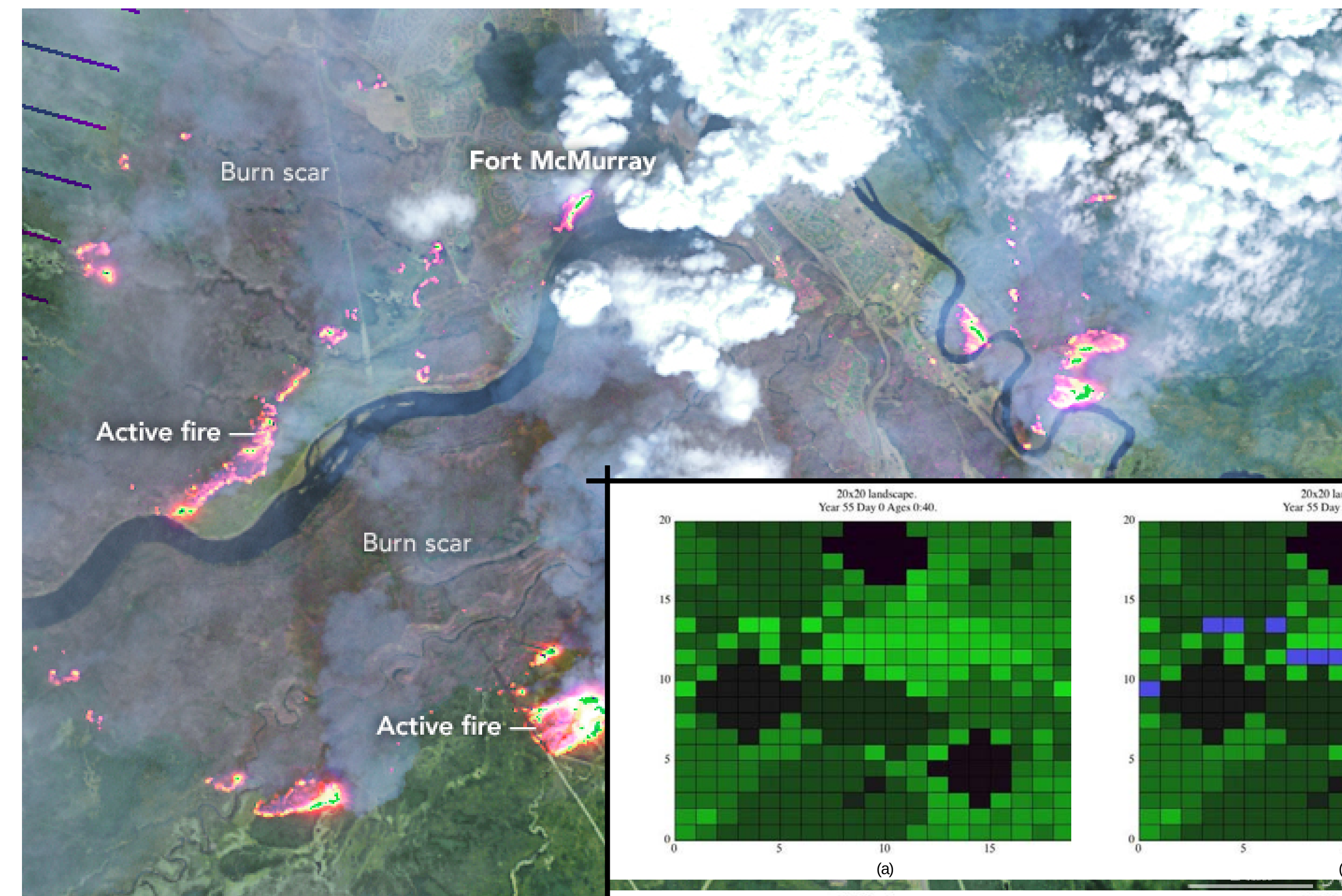
Spatially Spreading Systems

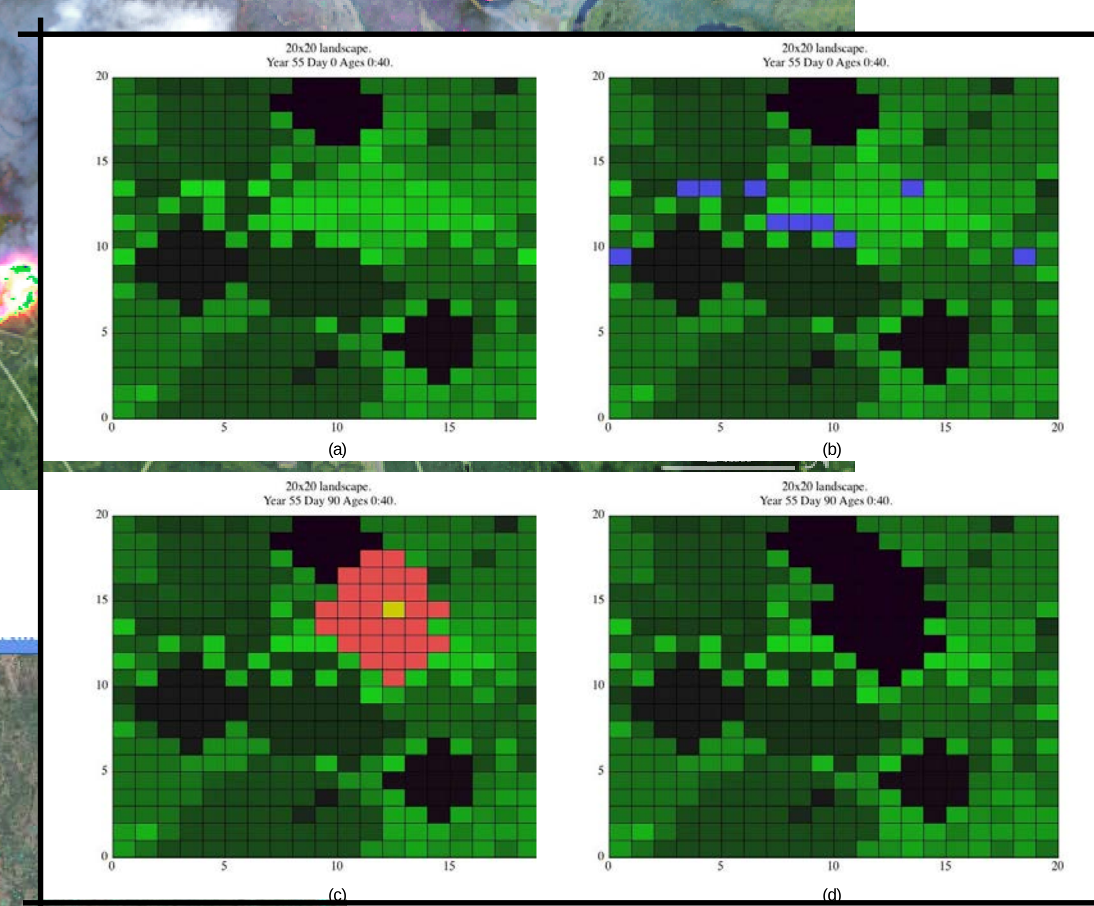
Invasive Species Control



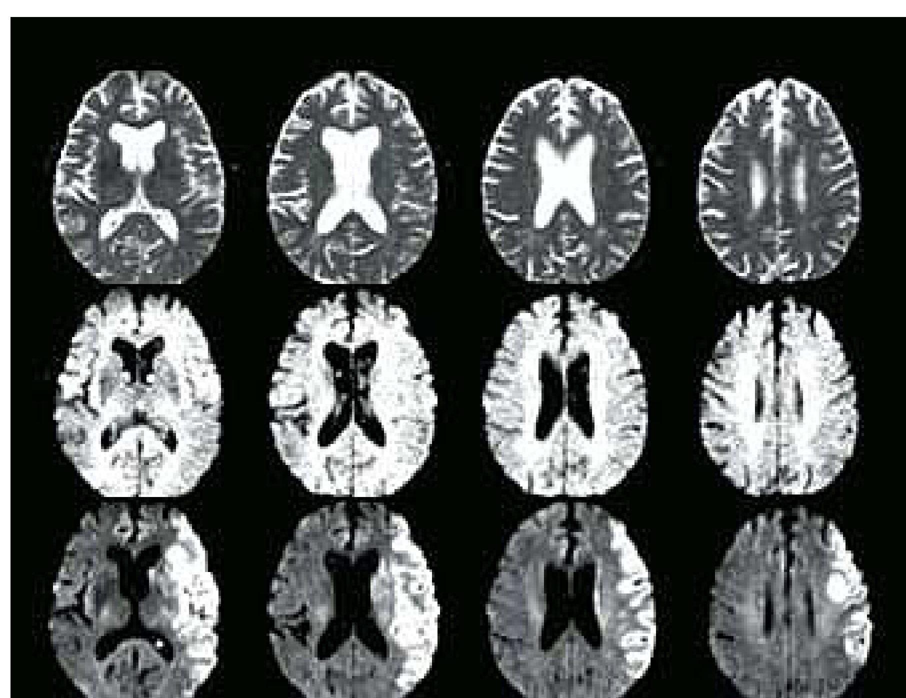
Sustainable Forest Management

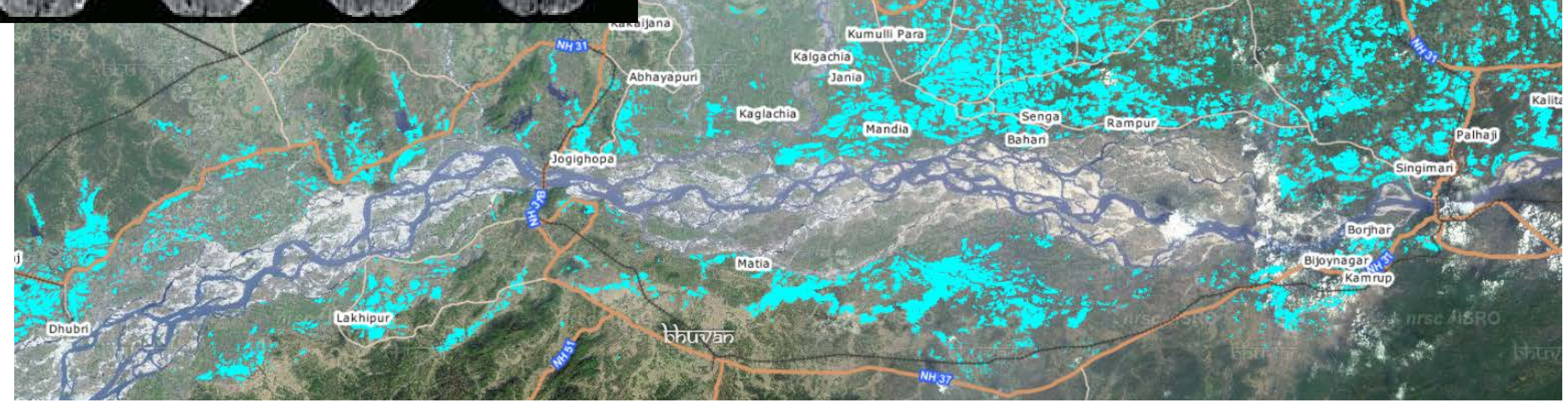
Predicting and Preventing Forest Wildfires





Medical Imaging





Flood Prediction

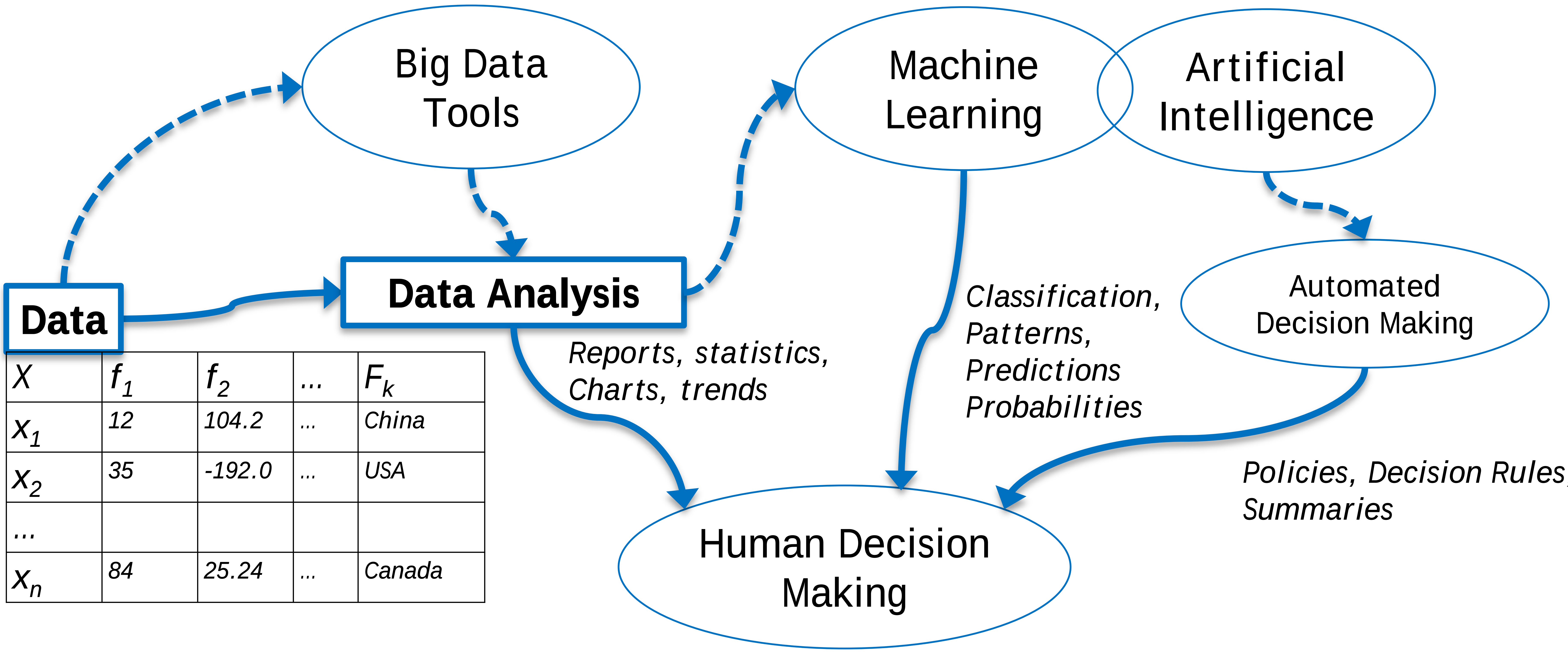
My AI/ML Research on Spatially Spreading Systems

- Learn/Optimize Harvest policy target directly
  - **Equilibrium Policy Gradients for Forest Harvest Management**
- Learn a simplified version of dynamics through interaction/observation
  - PAC-MDPs on Invasive Species in River Networks
  - **Reinforcement Learning of an agent policy for fire spread**
  - Tune simulator parameters to match satellite images
- Deep Learning for Building Predictive/Descriptive Model from series of images
  - **Forest Fire : satellite or simulator images**
  - Diffusion MRI : model/classify/prediction diagnosis of Alzheimer's disease from dRMI brain scan data

Outline

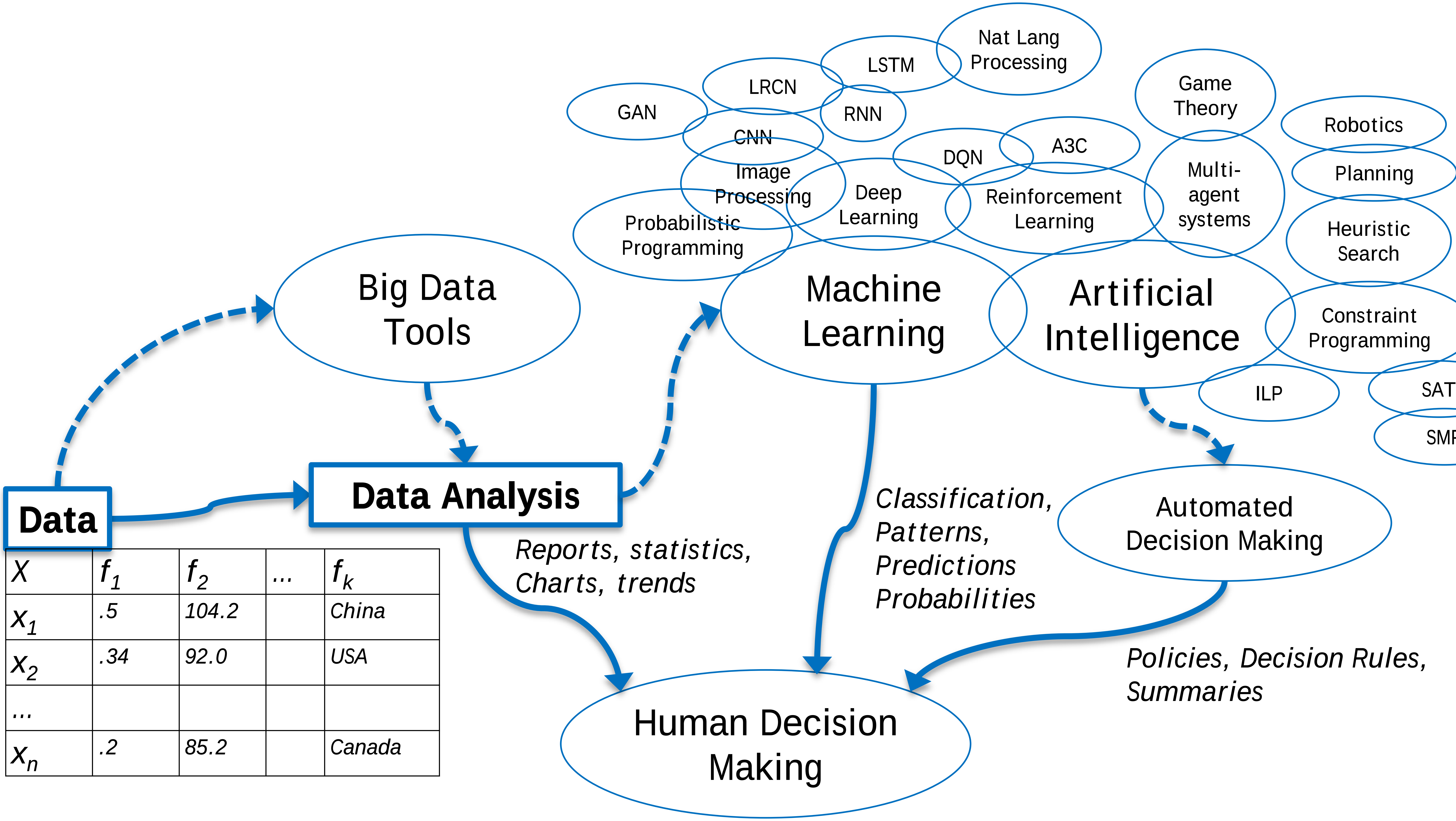
- My Interests
- Artificial Intelligence, Machine Learning, Big Data and all that...
- Decision Making / Reinforcement Learning
  - Sustainable Harvests, Let Burn Analysis, Fire Treatment Planning
- Deep Learning and Deep RL
  - Learning Spatial Dynamics for forest wildfire
    - Method I : Deep RL
    - Method II : LRCNs
- Future Challenges and Opportunities

Data, Big Data, Machine Learning, AI, etc, etc,...



X	f_1	f_2	...	F_k
x_1	12	104.2	...	China
x_2	35	-192.0	...	USA
...				
x_n	84	25.24	...	Canada

Data, Big Data, Machine Learning, AI, etc, etc,...

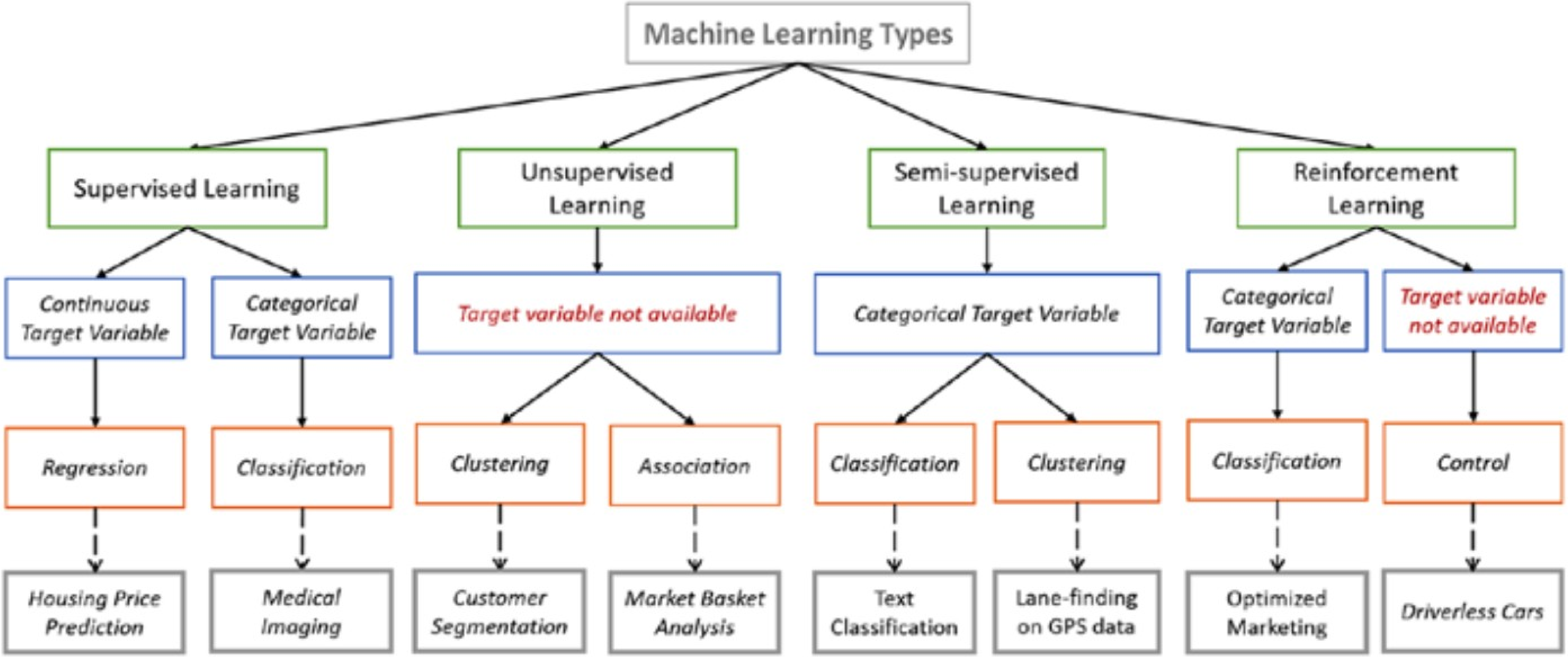


X	f_1	f_2	...	f_k
x_1	.5	104.2		China
x_2	.34	92.0		USA
...				
x_n	.2	85.2		Canada

Major Types of Machine Learning

"Detect patterns in data, use the uncovered patterns to predict future data or other outcomes of interest"

– Kevin Murphy, “Machine Learning: A Probabilistic Perspective”, 2012



Reinforcement Learning

A Markov Decision Process (MDP) is defined as:

States	$\mathbf{s} \in \mathbf{S}$	Start State s_0
Actions	$\mathbf{a} \in \mathbf{A}$	
Transition Dynamics	$\mathcal{T}(\mathbf{s}^t, \mathbf{a}^t, \mathbf{s}^{t+1}) : \mathbf{S} \times \mathbf{A} \rightarrow \delta(\mathbf{S})$	
Rewards	$r(\mathbf{s}^t, \mathbf{a}^t, \mathbf{s}^{t+1}) : \mathbf{S} \times \mathbf{A} \times \mathbf{S} \rightarrow \Re$	
Policy	$\pi(\mathbf{a} \vert \mathbf{s}) : \mathbf{S} \rightarrow \delta(\mathbf{A})$	

Maximize Expected Value:

$$V^\pi(s_0) = \mathbb{E}_\pi \left[r_0 + \gamma r_1 + \gamma^2 r_2 + \ldots \right]$$

Reinforcement Learning as an MDP

- RL [Sutton and Barto 1998]
  Approximate Dynamic Programming (ADP) [Powell 2007, 2009]
- Reinforcement Learning is learning the policy $\pi(a|s)$ for an MDP
- But without having access to the full definition of
  - the rewards $r(s^t, a^t, s^{t+1})$
  - AND/OR the dynamics $\mathrm{T}(s^t, a^t, s^{t+1})$
- Training data must be requested interactively, commit to action to find out the next state and reward.

$$\boxed{V^*(s) = R(s) + max_a \gamma \sum_{s'} P(s'|s,a) V^* s'}$$

Two Part Spatial Policy

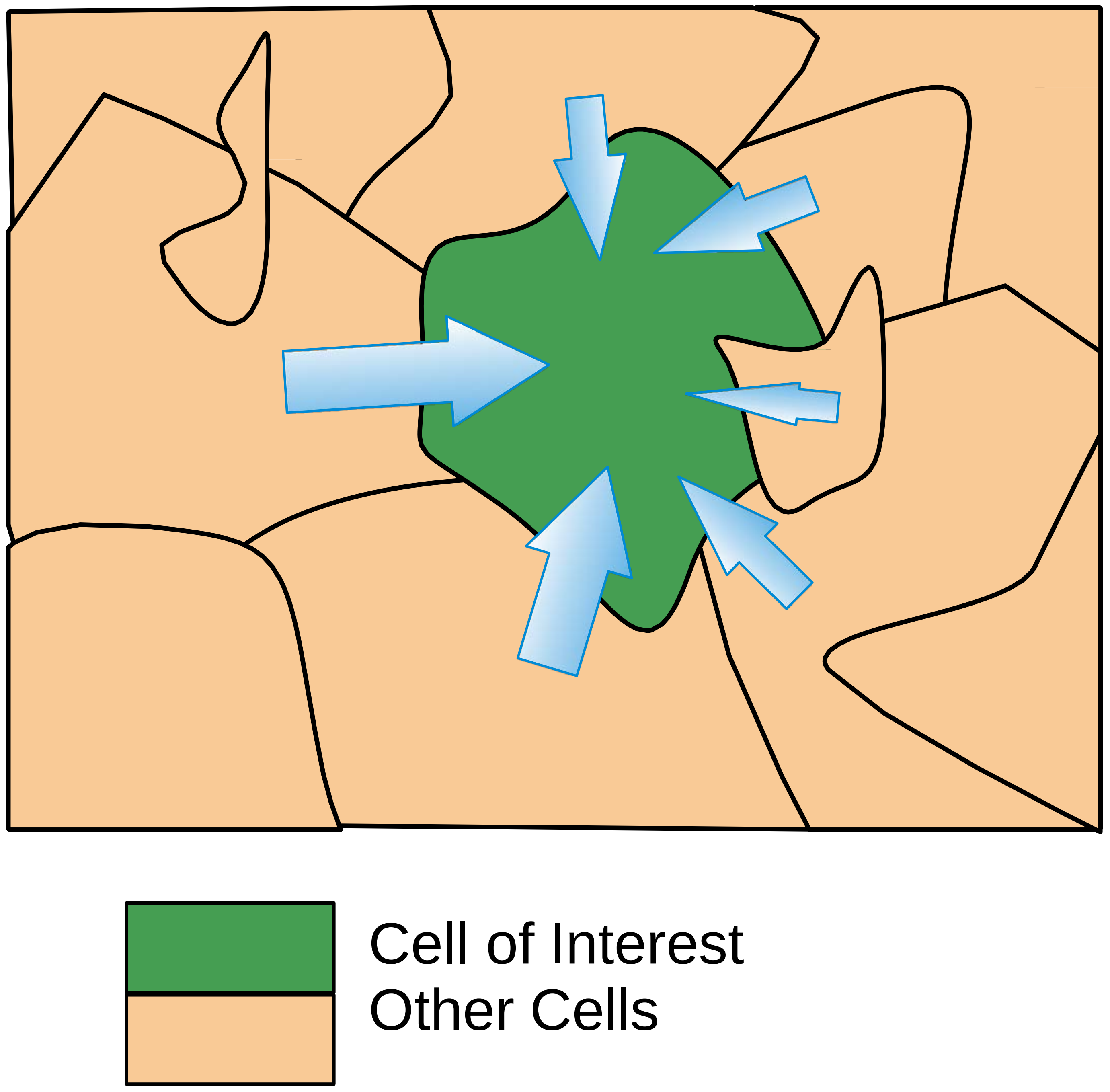
- **Cell Policy:**
  - Distribution over actions for a single cell given local state as well as other relevant states and actions elsewhere
- **Landscape Policy:**
  - Distribution over landscape actions given states of all cells

Cell Policy

$$\pi_c(a_c | a_{-c}, s, \theta) = \frac{\exp\left(\sum_f \theta_f(a_c) f_c(a_{-c}, s)\right)}{\sum_{b_c \in A} \exp\left(\sum_f \theta_f(b_c) f_c(b_{-c}, s)\right)}$$

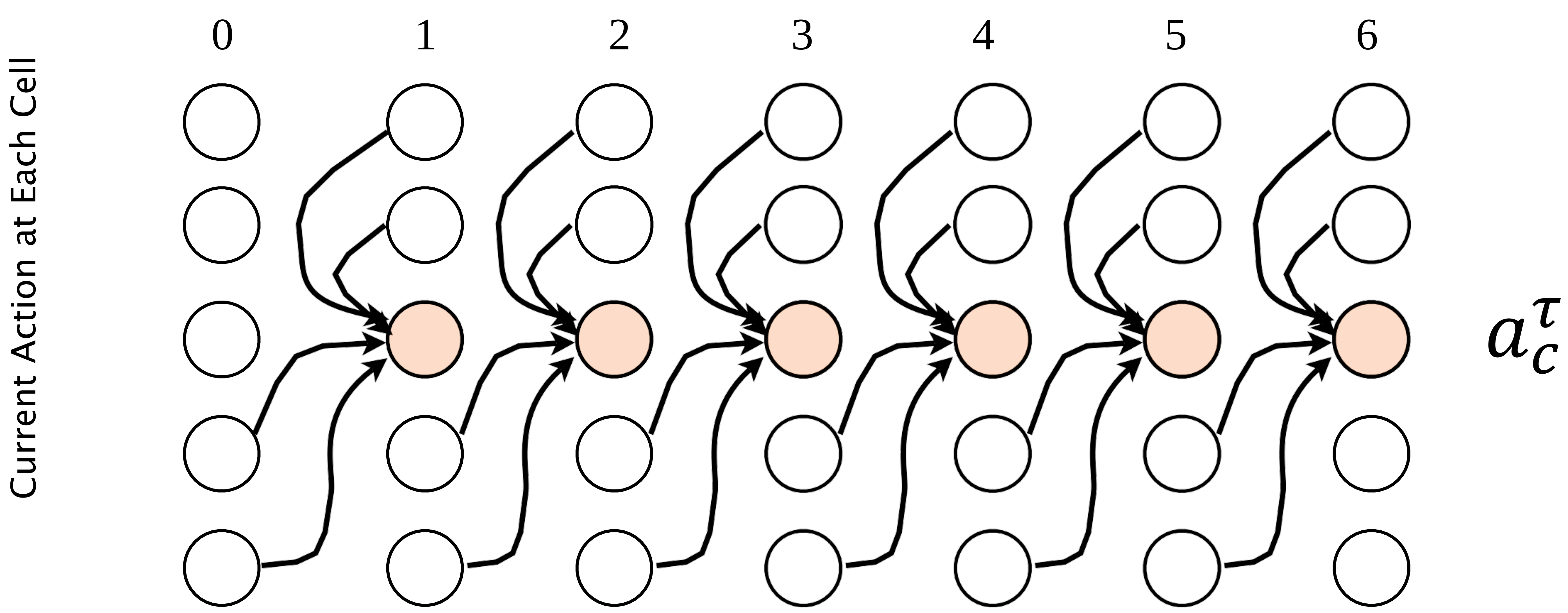
Cell Policy means:

What action should we take at this location if the actions at all other locations were already decided ?



Equilibrium Landscape Policy

A landscape policy $\pi_L(a|s,\theta)$ is the stationary distribution of a Markov chain defined by interacting with $\pi_c(a_c|a_{-c},s,\theta)$



- Obtaining an action from the policy requires inference, or in this case Gibbs sampling

(AAAI 2011, IEEE ToC 2014)

Policy Gradient Planning

- Gradient ascent in policy parameter space)
- Given initial or current policy parameters θ
- Collect trajectories K using current policy
- Estimate Gradient of Value of Policy
  (note that dynamics are not required)

$$\nabla_\theta V^\pi(\mathbf{s}_0) \approx \frac{1}{|K|} \sum_{\mathbf{k} \in K} R(\mathbf{k}) \sum_t \nabla_\theta \log \pi(\mathbf{a}^{\mathbf{k},t} | \mathbf{s}^{\mathbf{k},t}, \theta)$$

- Update Policy

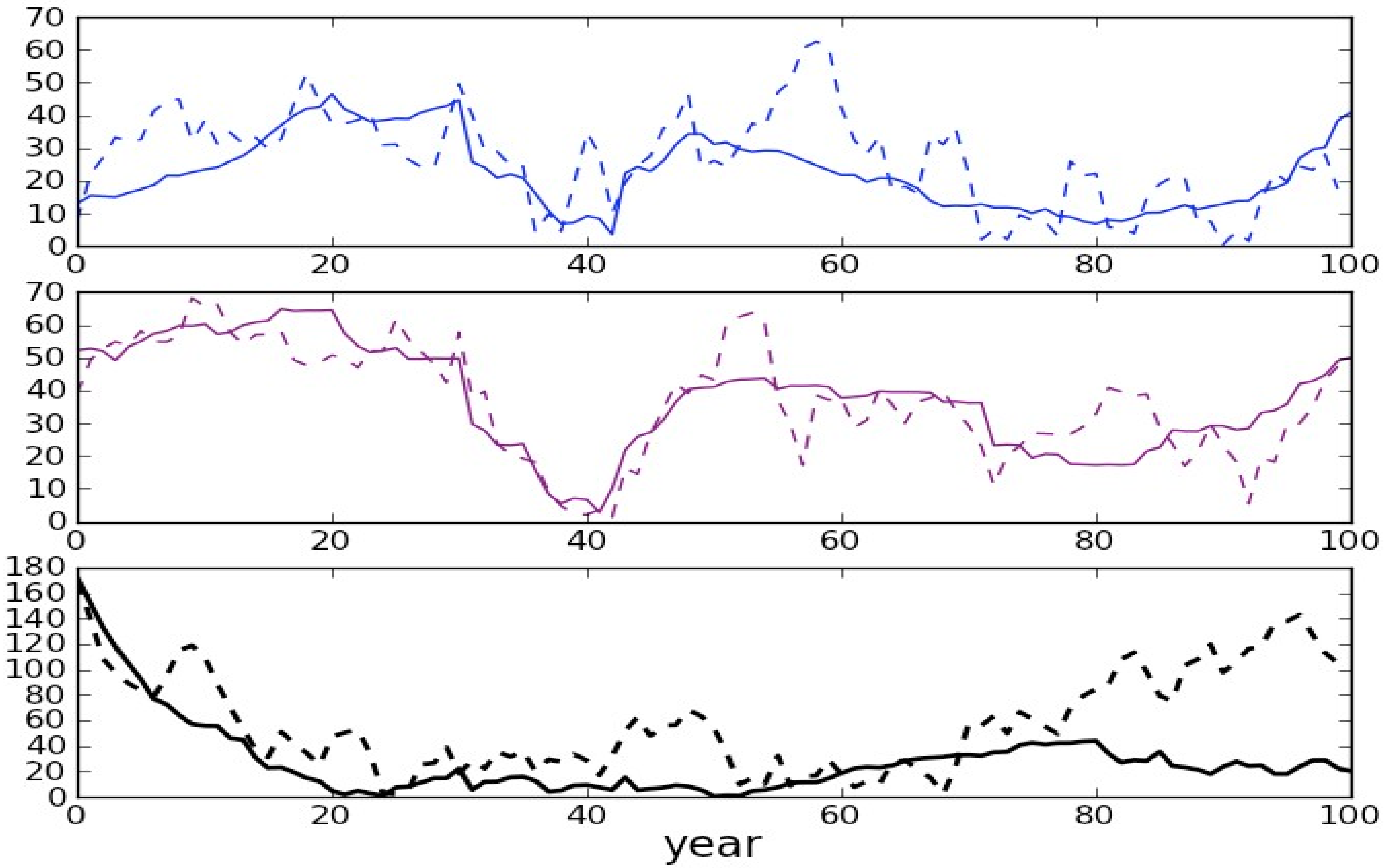
$$\theta' = \theta + \lambda \nabla_\theta \mathcal{V}$$

- Repeat until stopping condition

(AAAI 2011, IEEE ToC 2014)

Difference from Mean Total Forest Volume (10,000 km2)

Reward Models	
Harvest	Penalize Irregular Harvest
Total	Penalize Irregular Total Volume
Both	Penalize Both
All models reward total harvest volume and penalize adjacency violation	

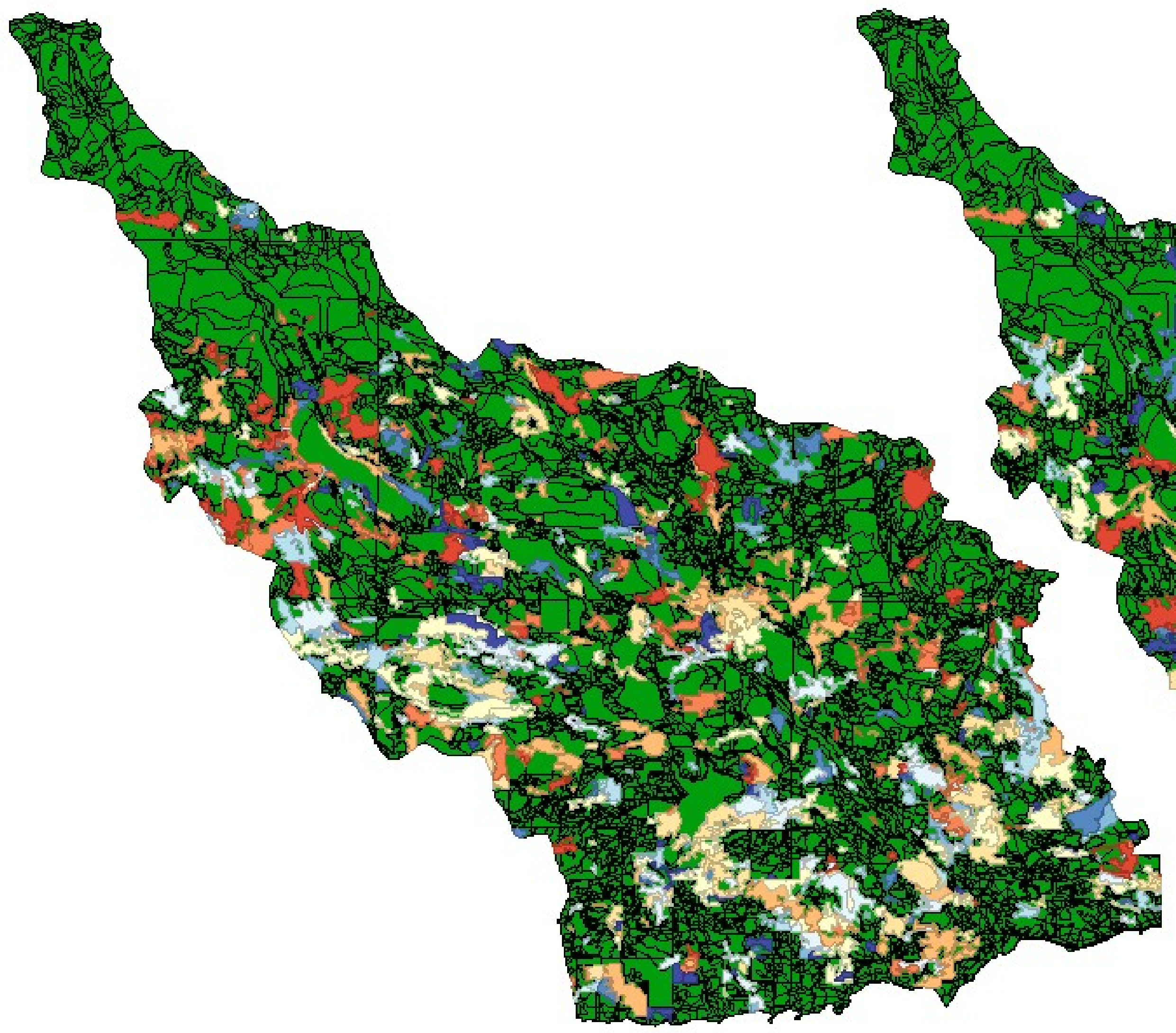


Forest Size Under Learned Policy

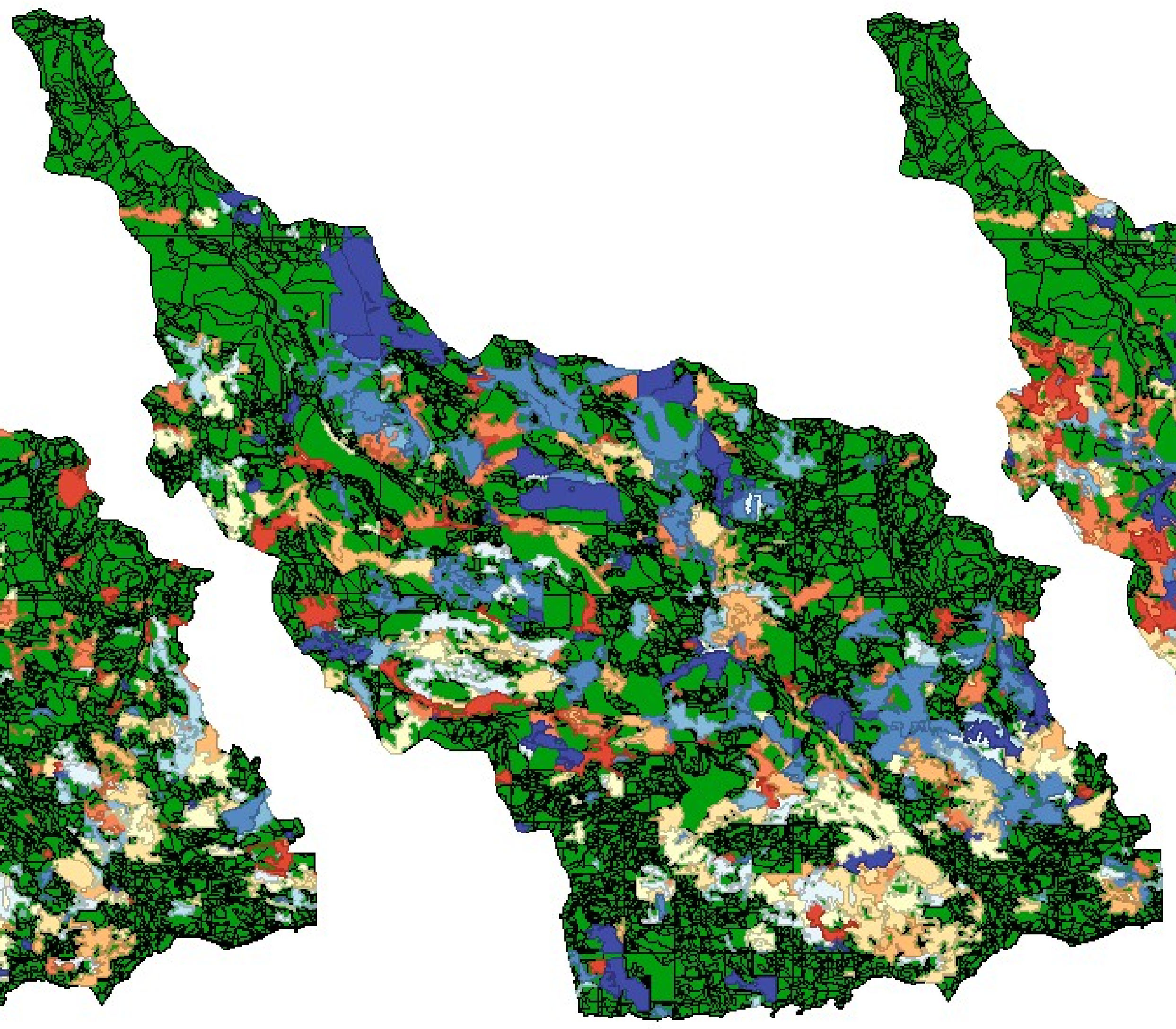
Forest Size Under Fixed Harvest mean

Three Example Plans

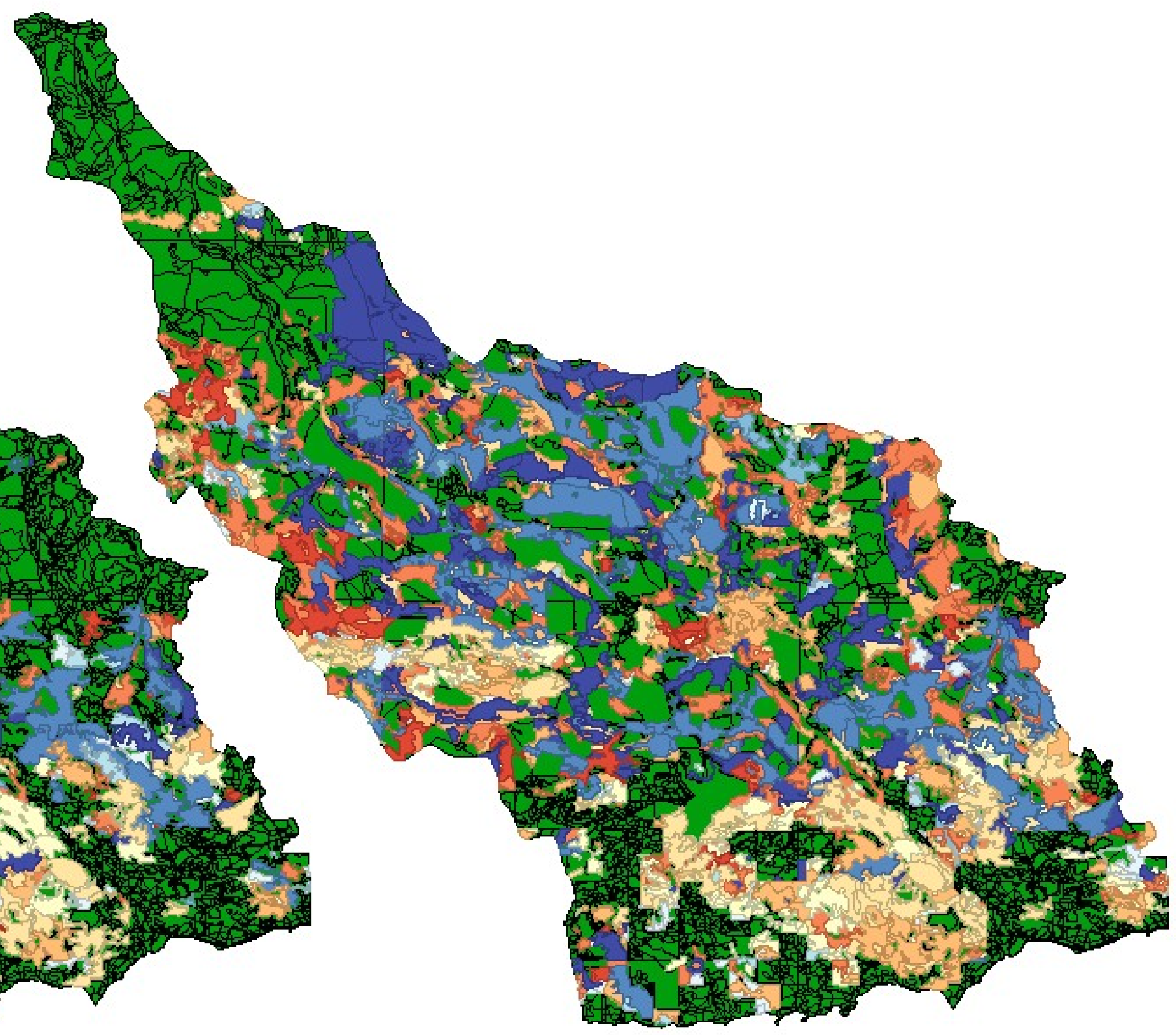
HAVR



HAVR High



AVR Collapsed



Typical results from HAVR reward model. Sustainable, low cut plan.

Common local minima from another run of HAVR. More aggressive plan, still sustainable over 100 years.

Unsustainable plan coming from an AVR run. Forest population collapses completely.

Decade in which cell was harvested

0-10	11-20	21-30	31-40	41-50	51-60	61-70	71-80	81-90	91-100

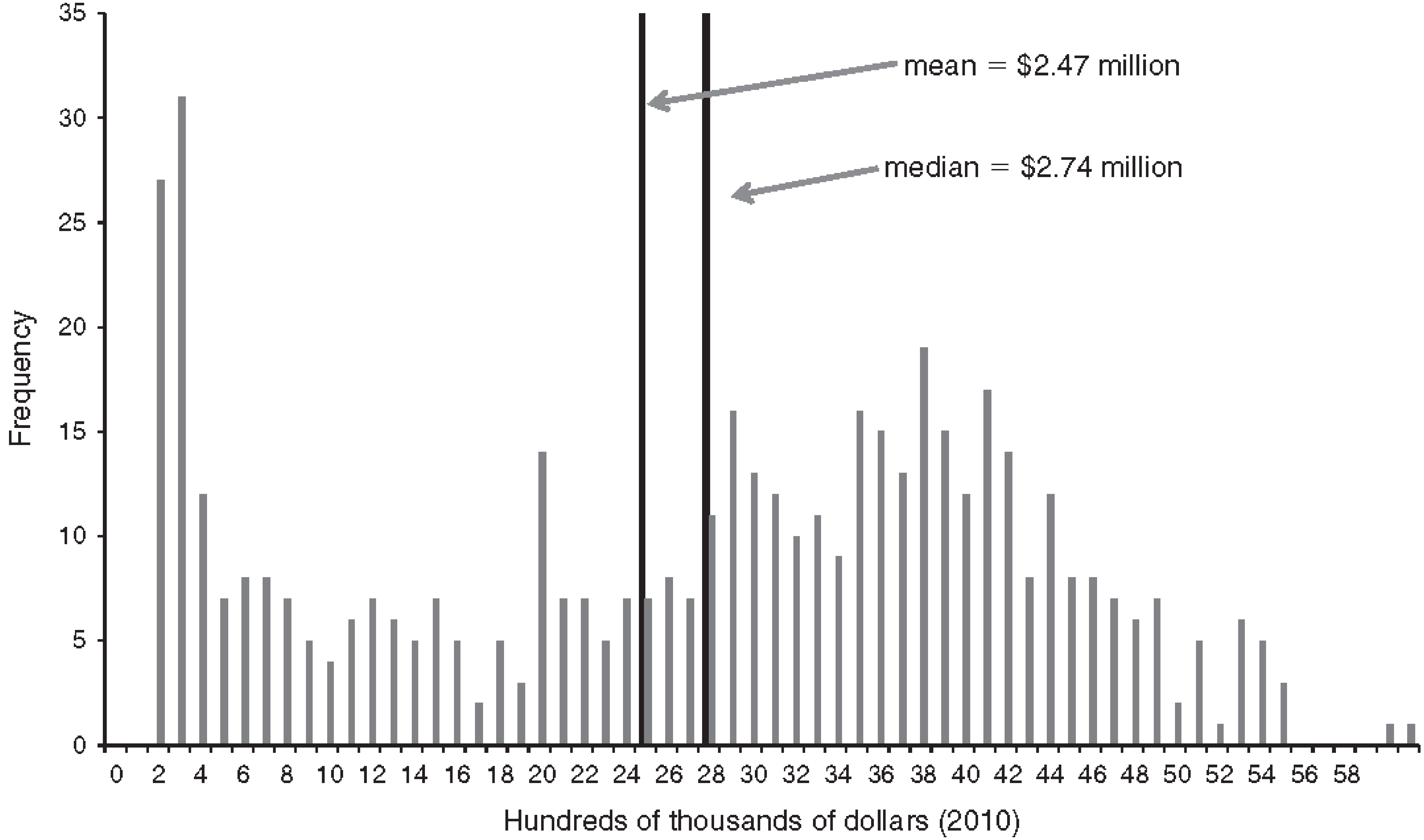
Forest Fire Simulator Platform

- While at OSU we created simulation framework [Houtman et al. (2013)]:
  - Combines a simple model of the spatial distribution of lightning strikes (based on historical data)
  - with the Farsite fire spread simulator [Finney, 1998]
  - a fire duration model [Finney et al.,2009]
  - and the high-resolution FVS forest growth simulator [Dixon, 2002].
  - Weather simulated by resampling from the historical weather time series observed at a nearby weather station.

OSU Farsite Based Simulation Framework

- Used this system for various results: descriptive, predictive and prescriptive analysis
  - Let-Burn analysis [Houtman et al. (2013)]
  - Economic Analysis of Fire suppression amongst multiple managers
  - Policy Optimization for Fire Suppression

Future Cost Savings for Let-Burn Policy



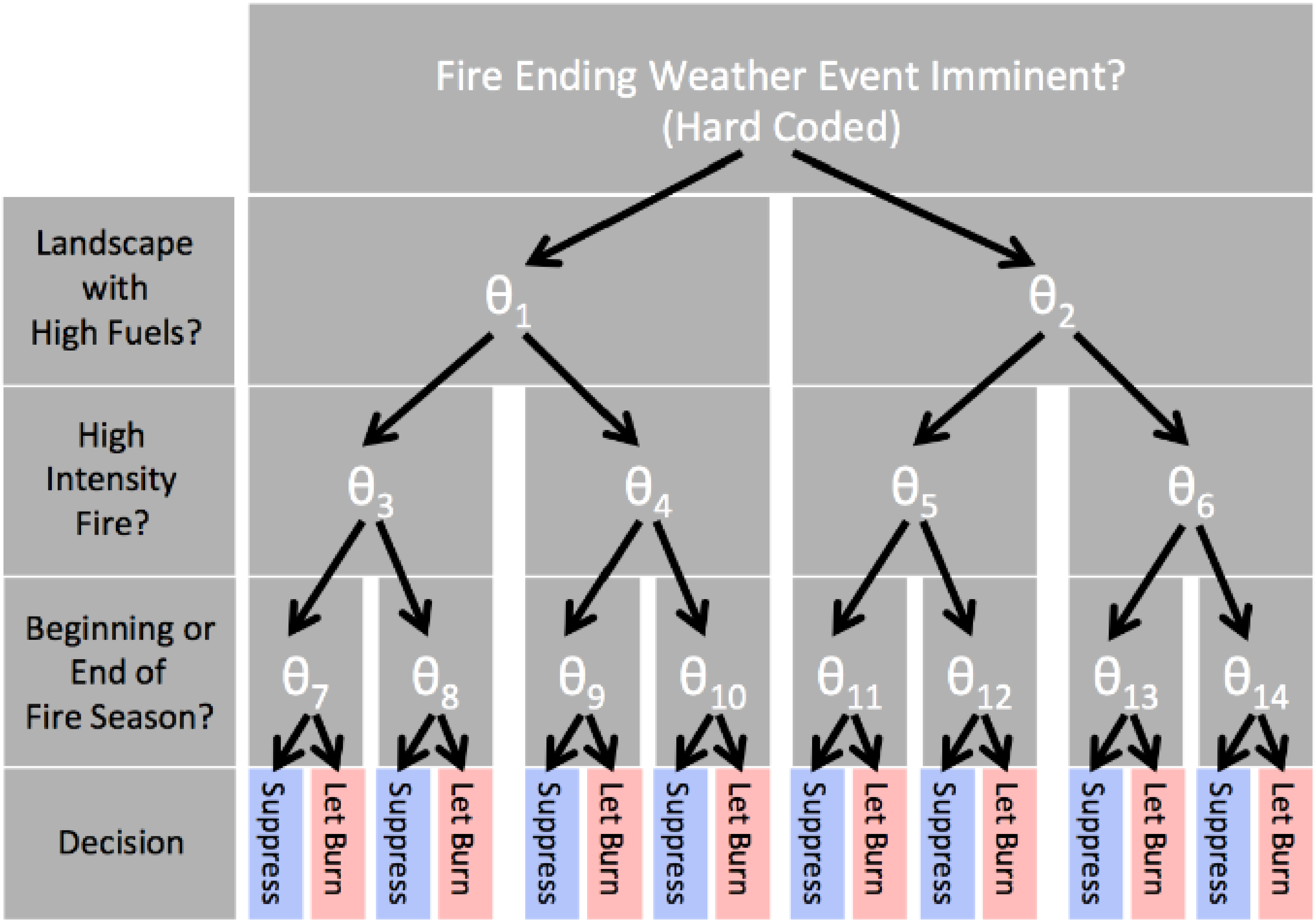
Back at OSU: Using ADP for Fire Treatment

- [Lauer 2017] use a stand based value function $Q_j(s, a)$ where j is each stand and choose the best action for that stand based on experience.
- In new work they also formulate a **multi-agent** (game theory) version of this where each agent is a different land manager and they operate on their own cells.
- The $Q_j^c(s, a)$ is now learned for each agent c and their actions are optimized as a Nash equilibrium.

Policy Optimization for Fire suppression

- [McGregor, 2017] Define fire suppression problem as an MDP, solve via direct policy search RL
  - Actions: suppress or let burn for given wildfire
  - Policy defined as function with parameters π_θ
- High resolution simulator : Farsite
  - Too slow for interaction → learn *surrogate model*
  - (MFMC) surrogate model learning – constructs new trajectories by pasting together parts of old ones
- Optimization of policy parameters
  - Sequential Model-based Optimization for general Algorithm Configuration (**SMAC**) [Hutter, 2010]
  - Iterative search for new θ values, model as random forest
- Multiple reward setups considered
  - Components – suppression cost, timber values, deviation from restoration target, air quality (burn days), recreation target

Parameterized Decision Tree Policy



[McGregor, 2017]

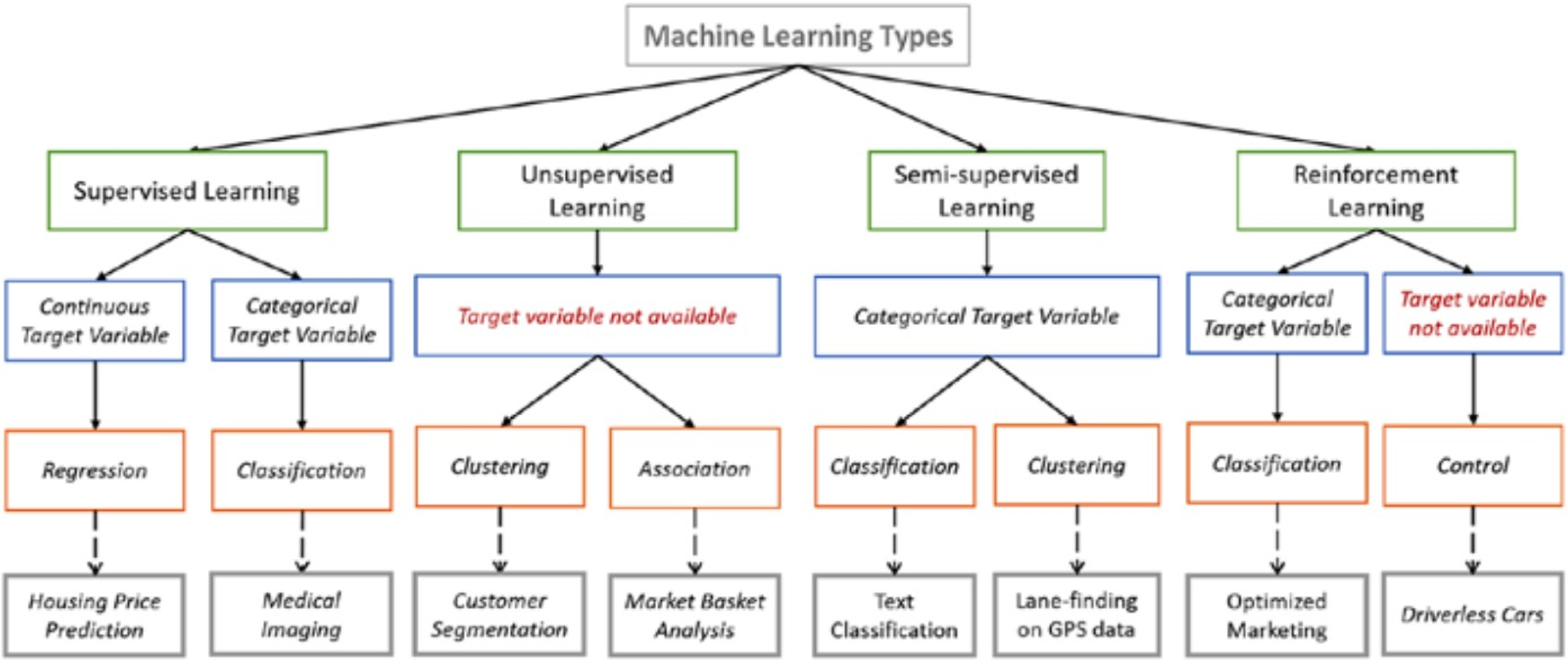
Outline

- My Interests
- Artificial Intelligence, Machine Learning, Big Data and all that...
- Decision Making / Reinforcement Learning
  - Sustainable Harvests, Let Burn Analysis, Fire Treatment Planning
- Deep Learning and Deep RL
  - Learning Spatial Dynamics for forest wildfire
    - Method I : Deep RL
    - Method II : LRCNs
- Future Challenges and Opportunities

Major Types of Machine Learning

"Detect patterns in data, use the uncovered patterns to predict future data or other outcomes of interest"

– Kevin Murphy, "Machine Learning: A Probabilistic Perspective", 2012



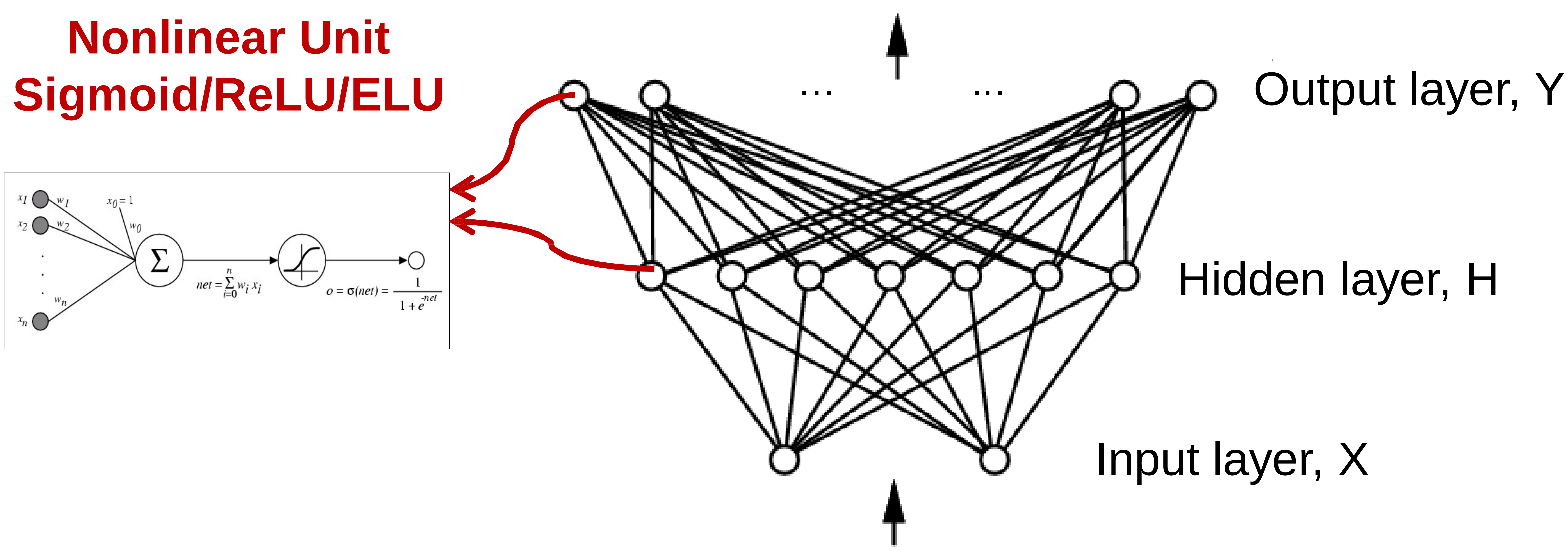
How do Neural Networks/Deep Learning Fits In?

- Deep Learning: methods which perform machine learning through the use of multilayer neural networks of some kind.
- Deep Learning can be applied in any of the three main types of ML:
  - **Supervised Learning** : very common, enormous improvement in recent years
  - **Unsupervised Learning** : just beginning, lots of potential
  - **Reinforcement Learning** : recent (past 3 years) this has exploded, especially for video games

Neural Networks to learn $f : X \rightarrow Y$

- f can be a non-linear function
- **X** (vector of) continuous and/or discrete variables
- **Y** (vector of) continuous and/or discrete variables

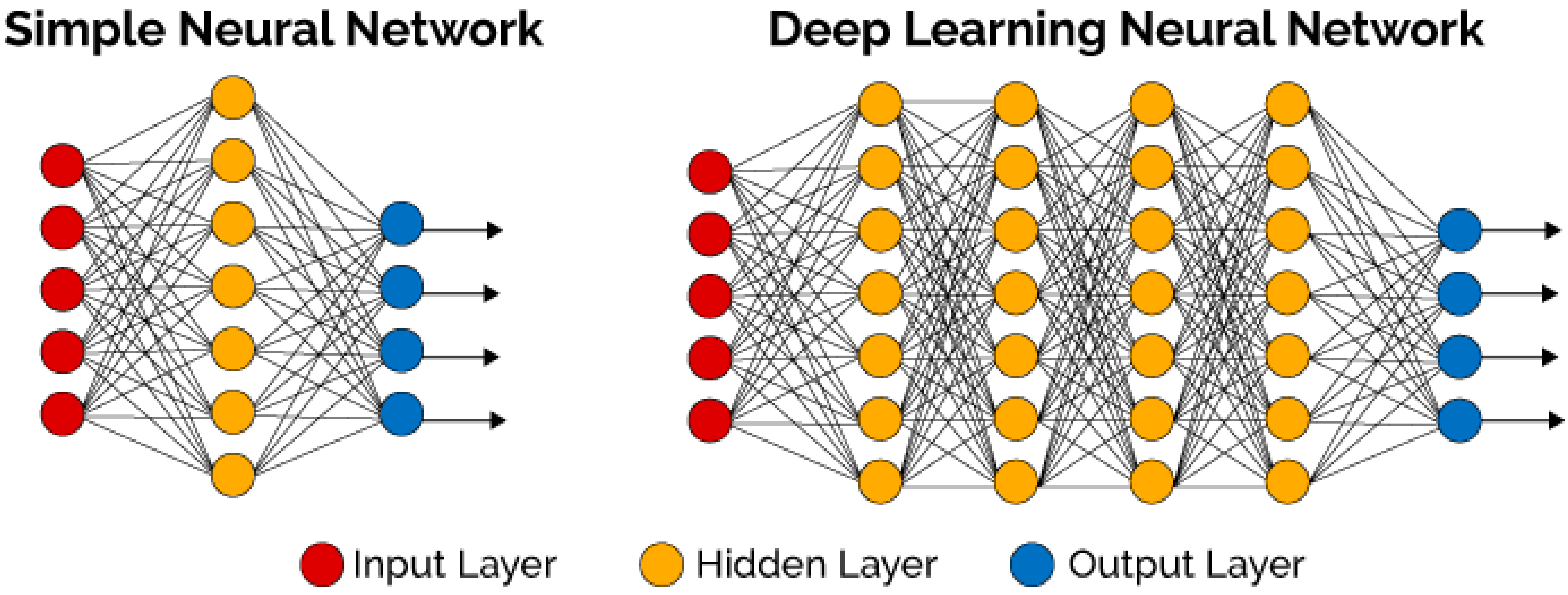
Feedforward Neural networks - Represent f by network of non-linear (logistic/sigmoid/ReLU) units:



Basic Three Layer Neural Network

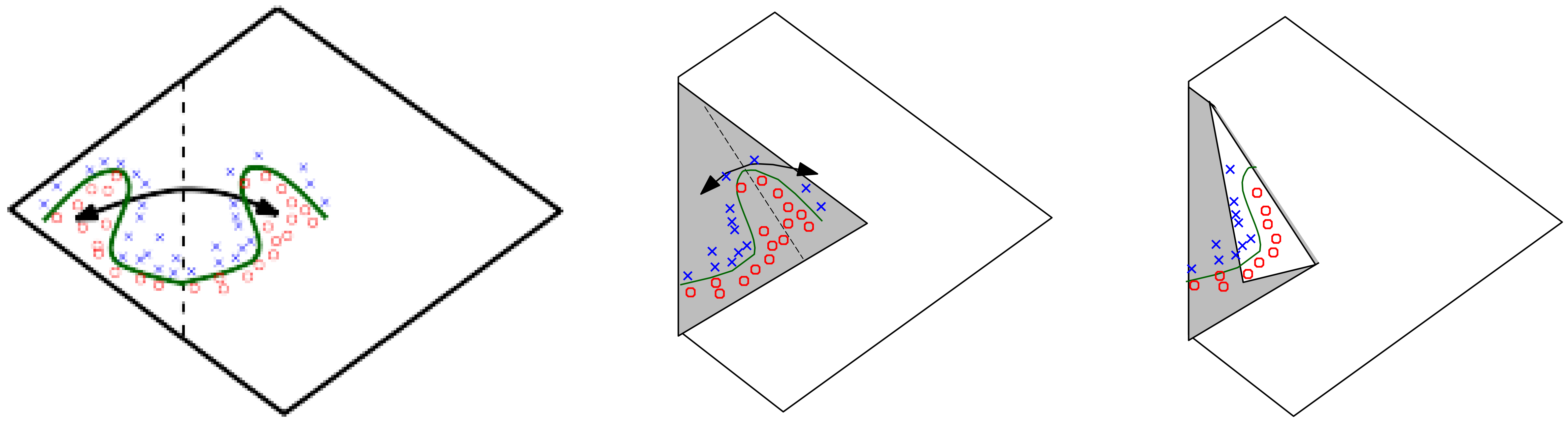
- Input Layer
  - vector data, each input collects **one feature**/dimension of the data and passes it on to the (first) hidden layer.
- Hidden Layer
  - Each hidden unit computes a weighted sum of all the units from the input layer (or any previous layer) and passes it through a **nonlinear activation function**.
- Output Layer
  - Each output unit computes a weighted sum of all the hidden units and passes it through a (possibly nonlinear) **threshold function**.

So What is Deep Learning?



- Hackernoon

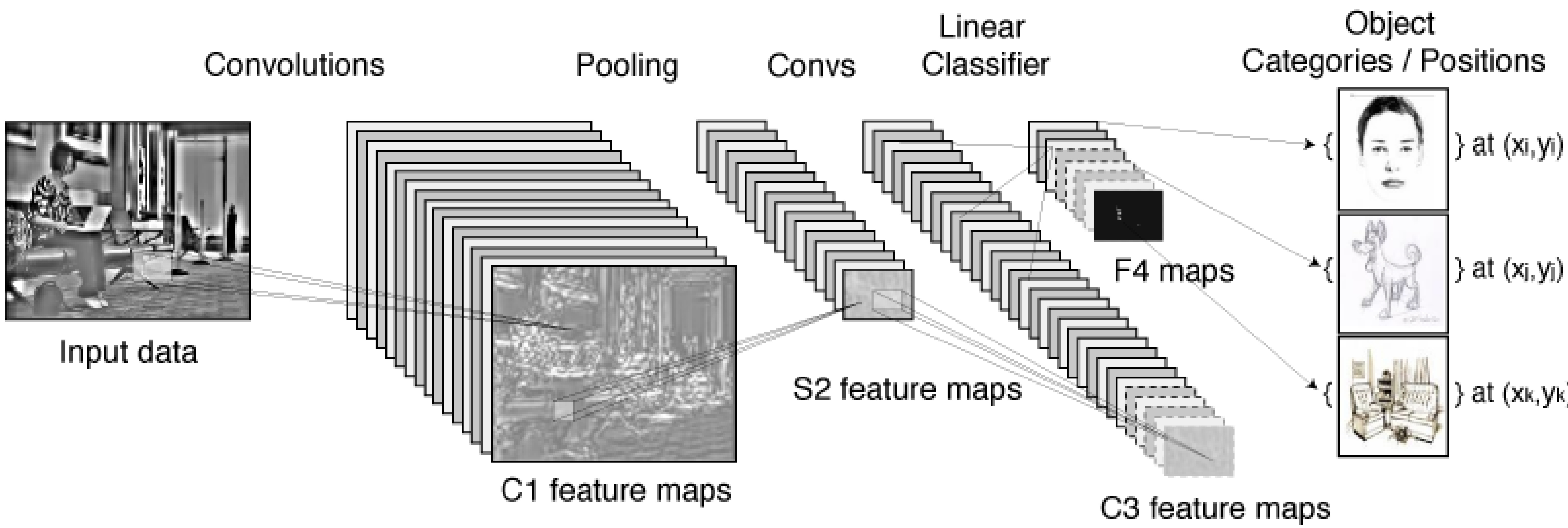
Why Go Deep?



(Goodfellow 2016)

Using Hidden LeRU units, each hidden layer increases power, exponential advantage of additional layers.

Convolutional Neural Networks (CNNs)

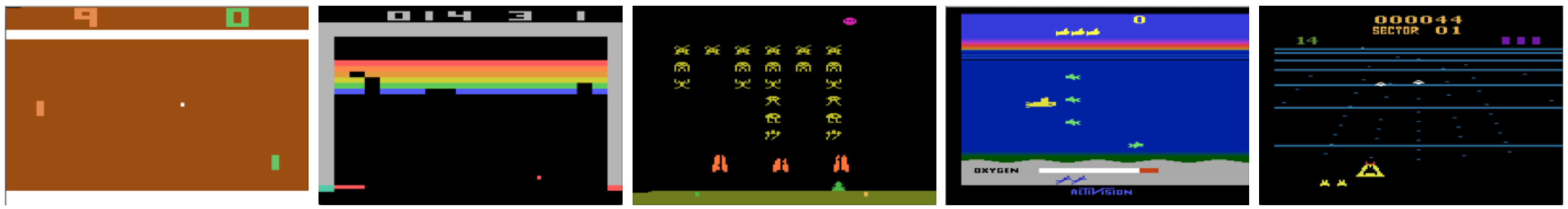


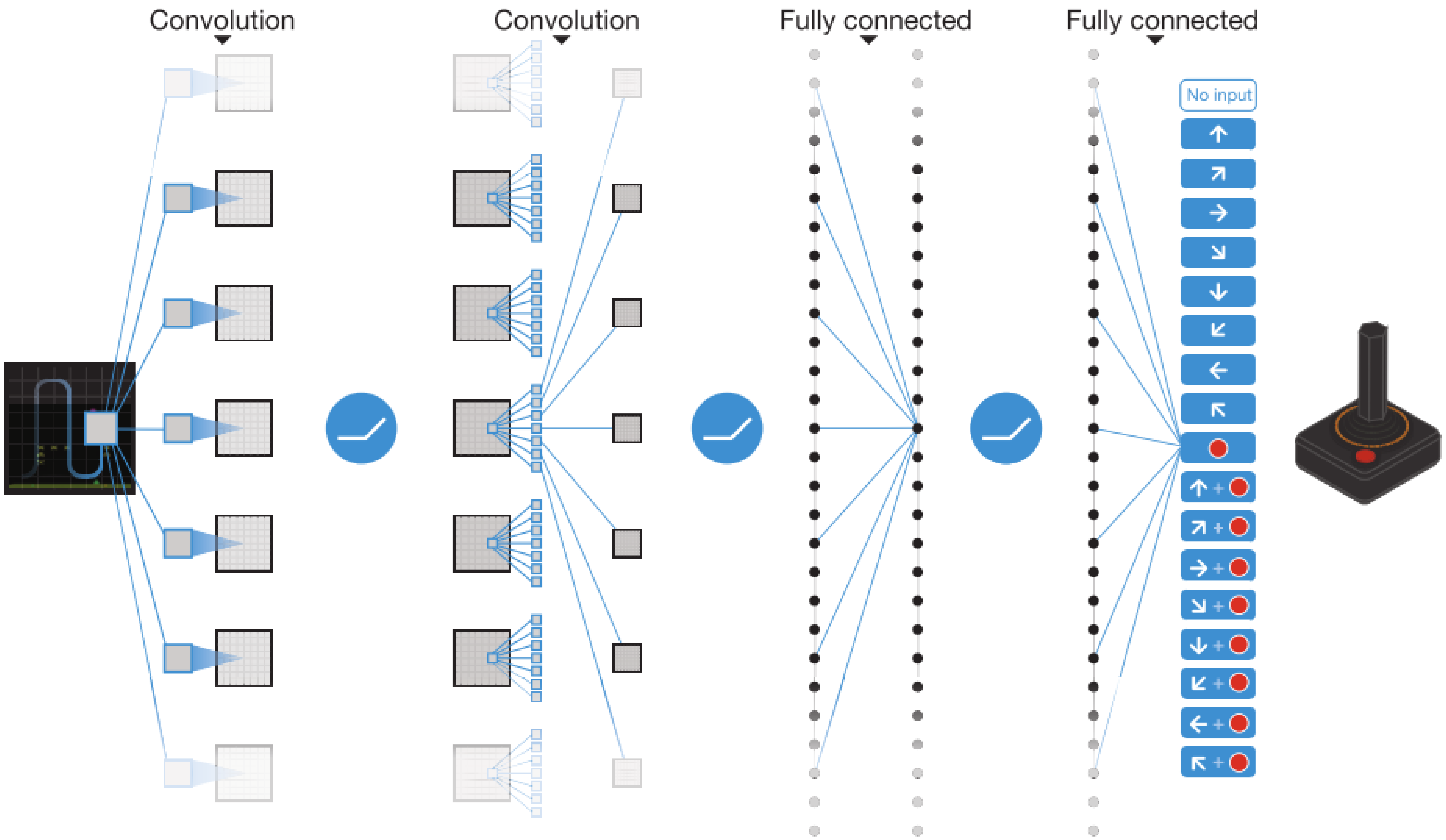
Deep RL: Deep Reinforcement Learning

- CNNs + Fully Connected Deep Network for learning a representation of a policy
- Reinforcement Learning for updating the policy through experience to make improved decision decisions Requires a value/reward function

Deep Learning, Deep RL, DQN, A3C, etc...

- Recent flurry of advances by Google DeepMind and others applying Deep Learning to RL algorithms
- Deep Q-Learning – **DQN**
- Asynchronous Advantage Actor-Critic - **A3C**





Outline

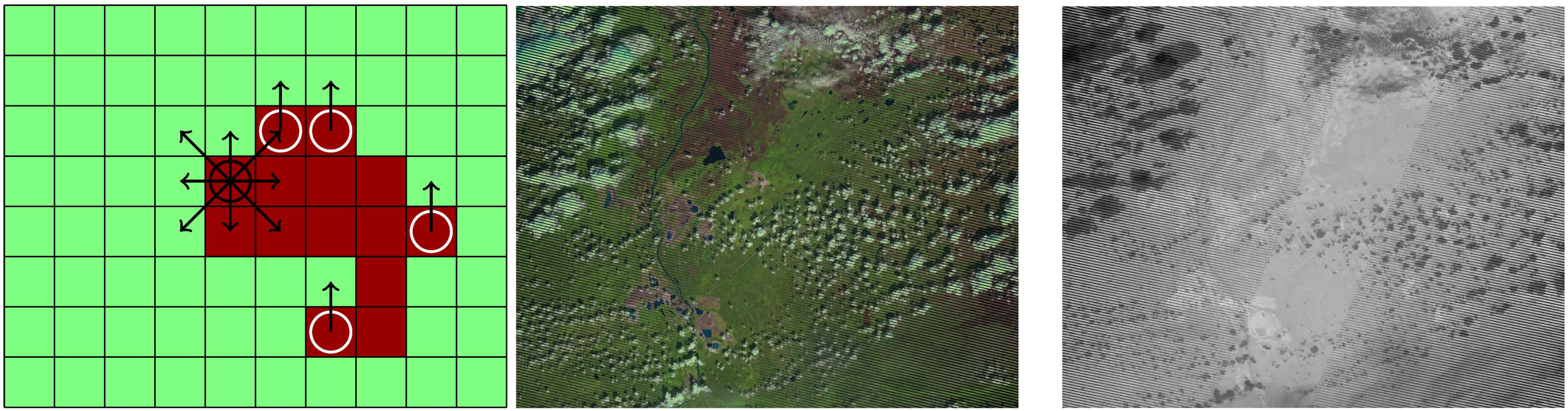
- My Interests
- Artificial Intelligence, Machine Learning, Big Data and all that...
- Decision Making / Reinforcement Learning
  - Sustainable Harvests, Let Burn Analysis, Fire Treatment Planning
- Deep Learning and Deep RL
  - Learning Spatial Dynamics for forest wildfire
    - Method I : Deep RL
    - Method II : LRCNs
- Future Challenges and Opportunities

Machine Learning for Forest Fire Modelling

- Suppression:
  - Letting Fires Burn: given a fire that is occurring, decide to let it burn or not
  - Trade off between suppression costs, cost of large fires, benefit of free fuel reduction on reducing future fires.
- Sustainable Harvests:
  - find harvest schedule maximizing value while satisfying spatial constraints each year
  - sustainability constraints over many years.
- Fuel Treatment:
  - find optimal treatment of fuels over time to reduce expected cost of catastrophic fires.
- Fire Spread Dynamics:
  - Imagine fire is the agent on the landscape, each spread from cell to cell is an action.
  - Learn a fire spread policy from data

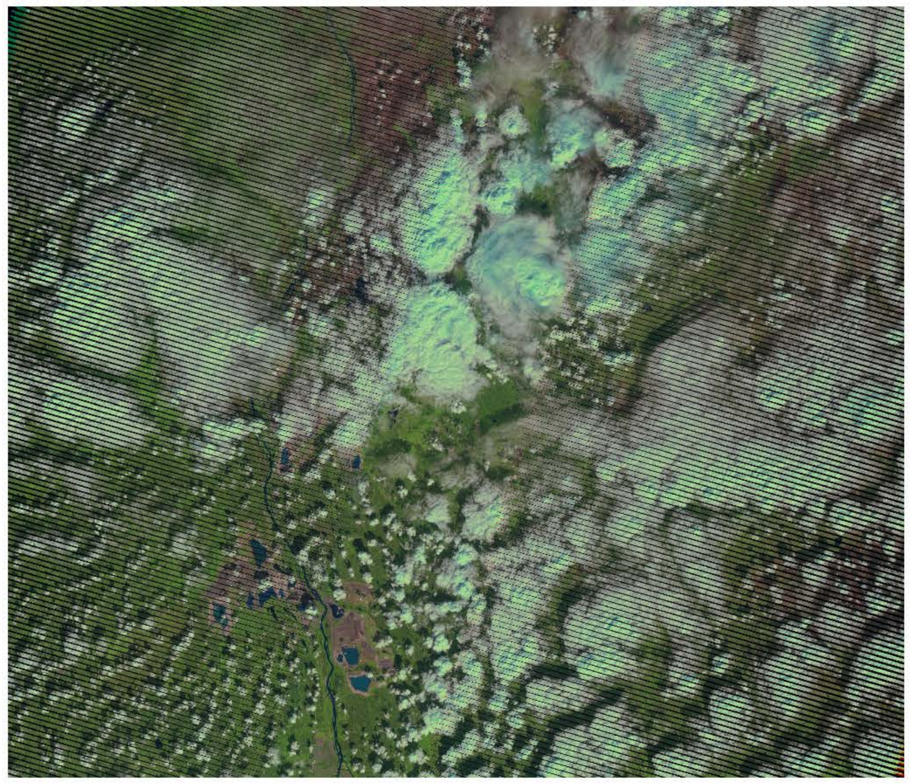
Method I: Problem setup

- Using satellite images from two large forest wildfires in Northern Alberta : Richardson 2011, For McMurray 2016.

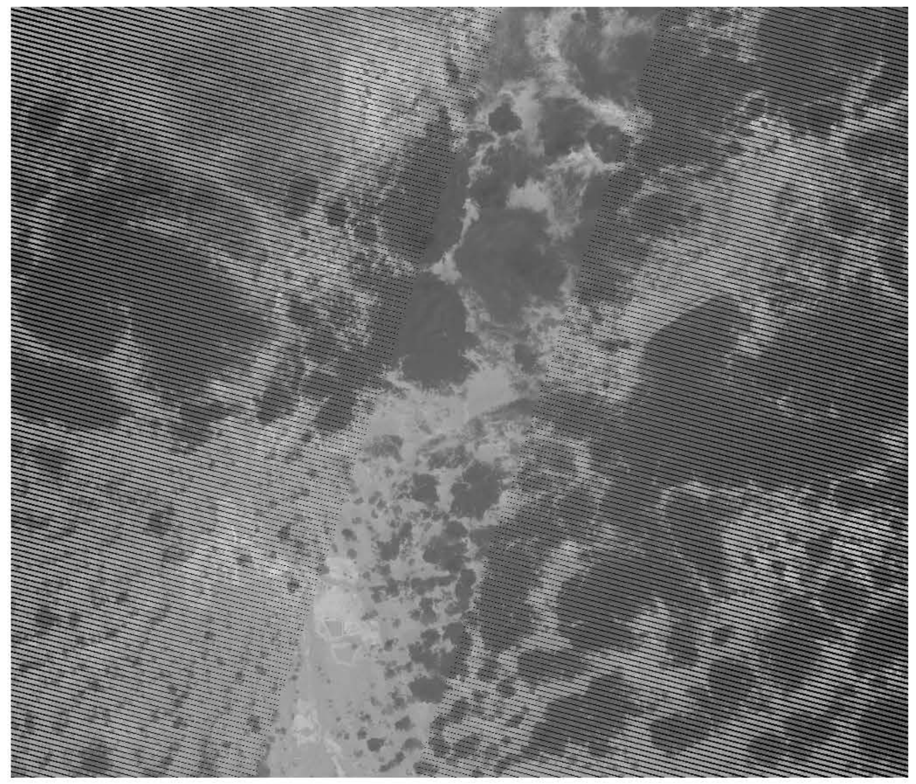


(a) Schematic of the state and actions (b) Raw Color Image (c) Thermal Image

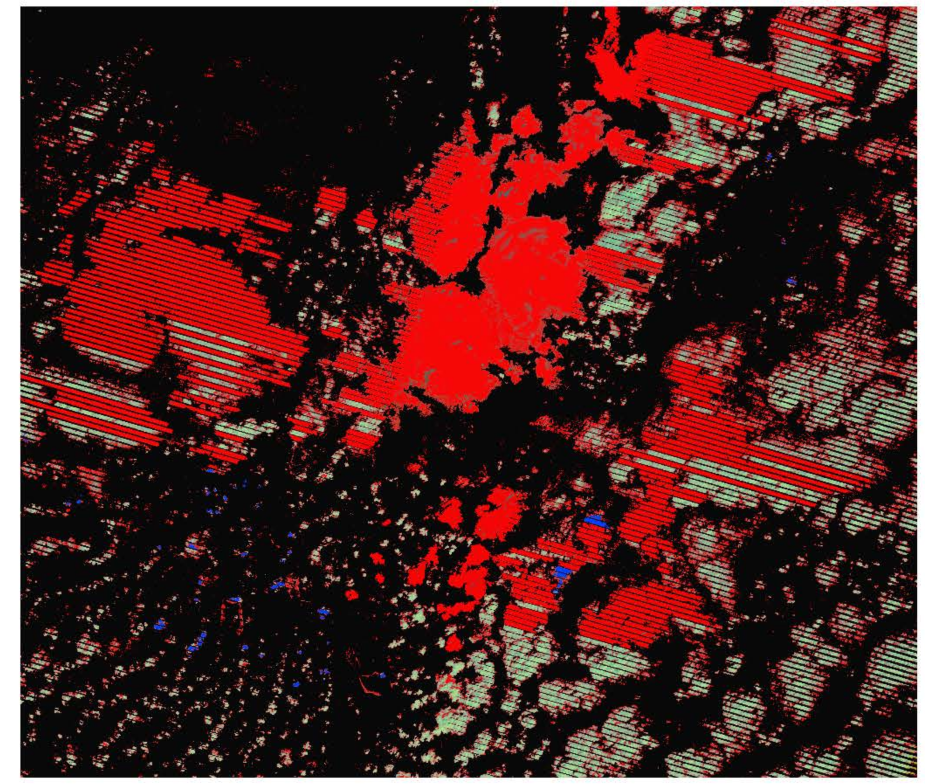
Results



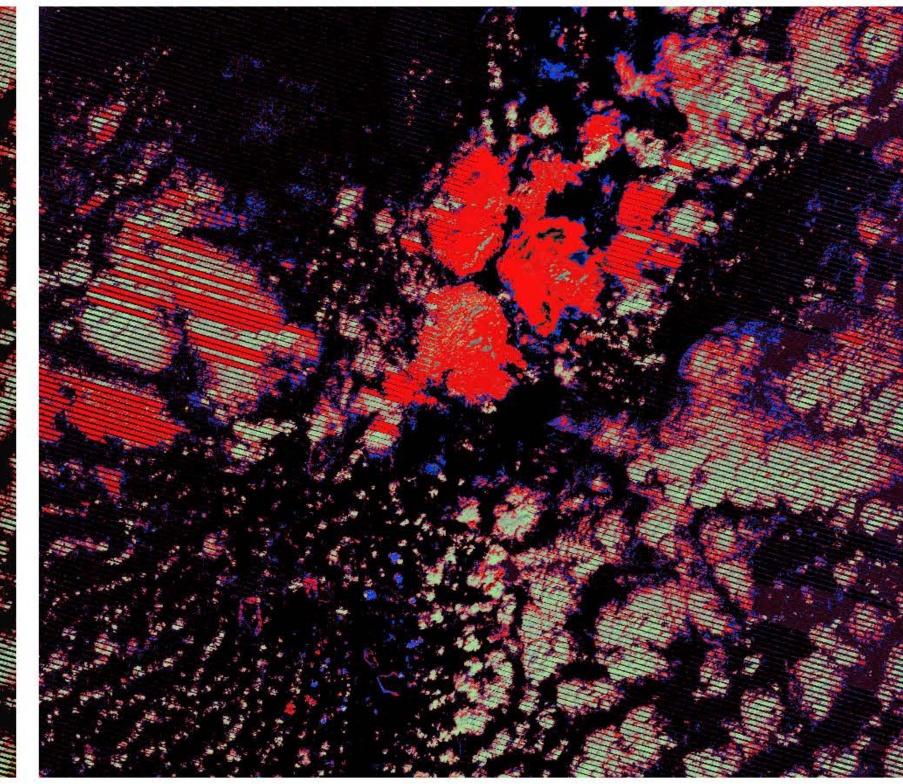
(a) Satellite Image of August 11



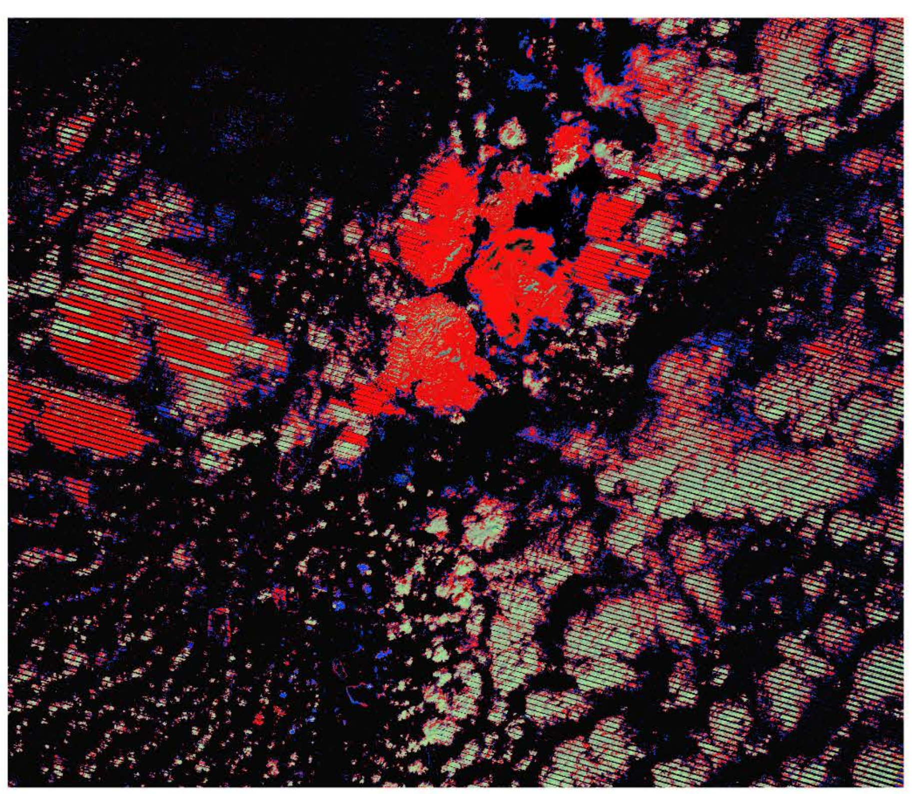
(b) Thermal Image of August 11



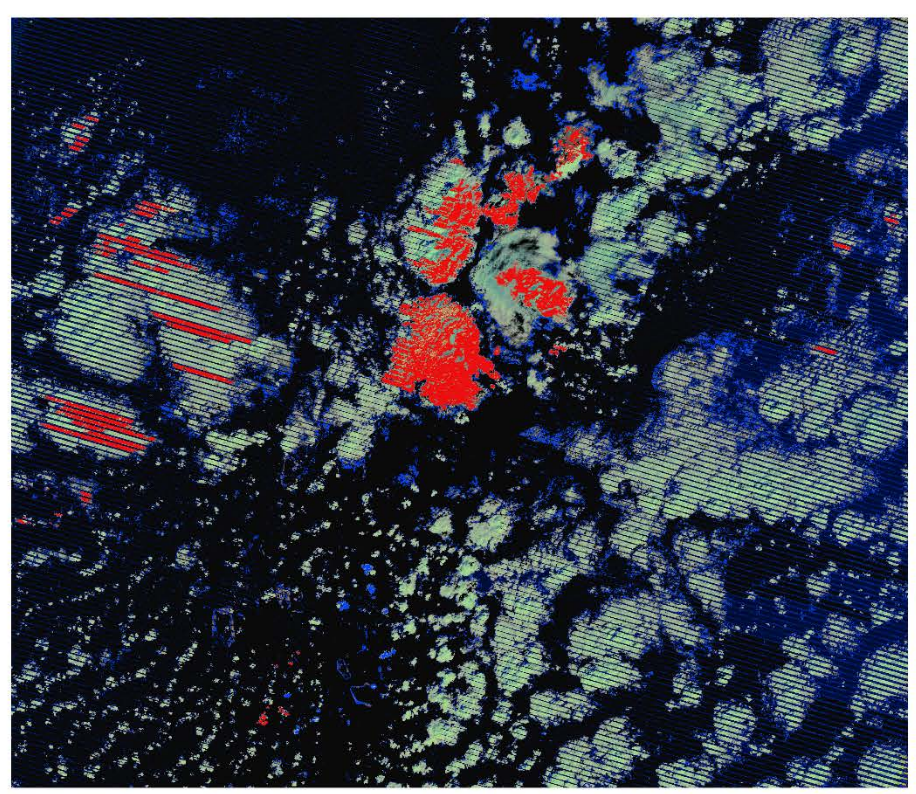
(c) Gaussian Process



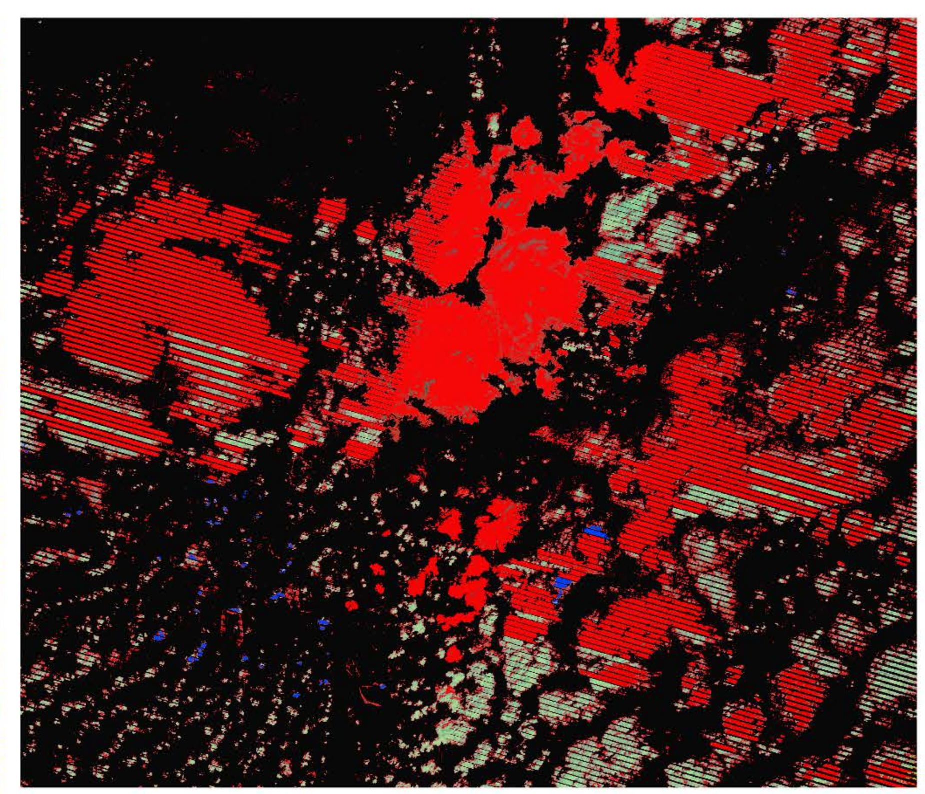
(d) Value Iteration



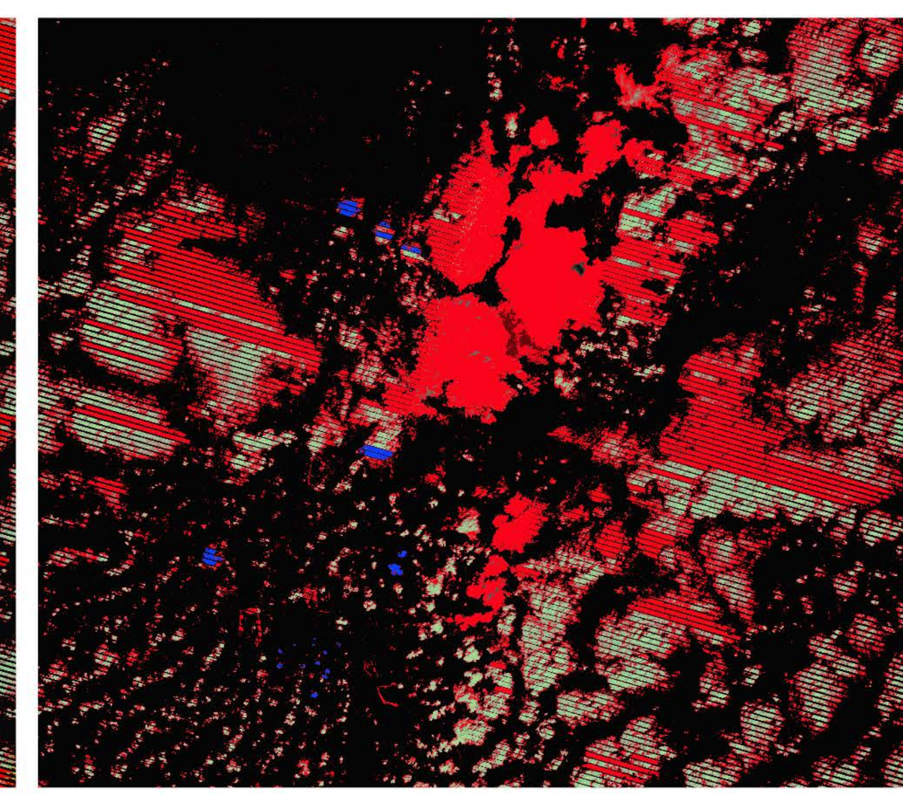
(e) Policy Iteration



(f) Q Learning



(g) MCTS

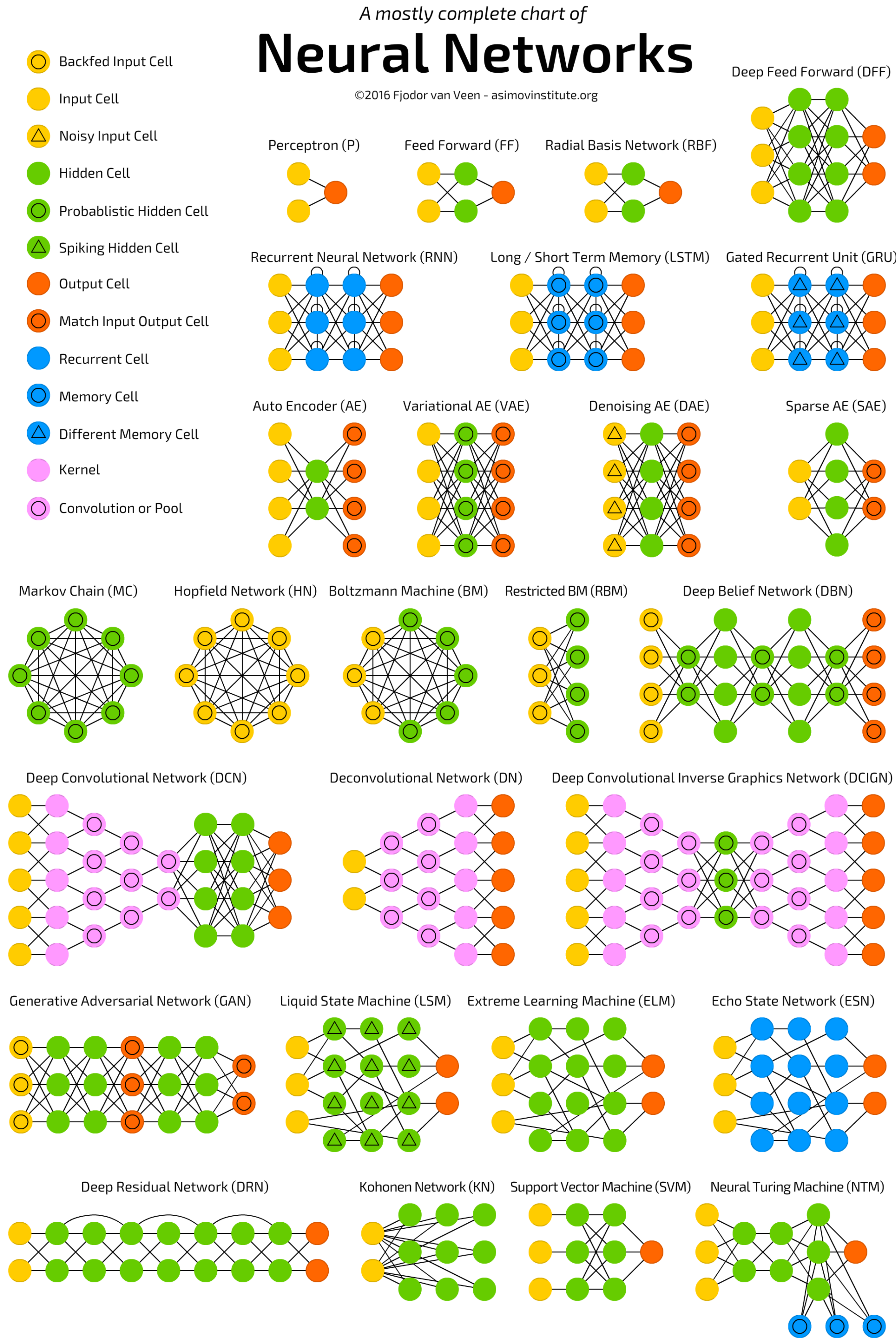


(h) A3C

Results

Method	(C)	(D)	(E)	(F)
GP	60.5%	47.9%	45.3%	**20.5** %
V.I	88.5%	68.4%	30.1%	6.4%
P.I	89.3%	67.8%	35.8%	8.9%
Q.L	84.2%	61.4%	26.4%	5.3%
MCTS	65.3%	55.7%	49.7%	5.8%
A3C	**90.1**%	**81.8**%	**50.8**%	13.4%

Table 3: Average Accuracy of each algorithm trained on the Richardson Fire but applied on the Fort McMurray fire for different time durations.



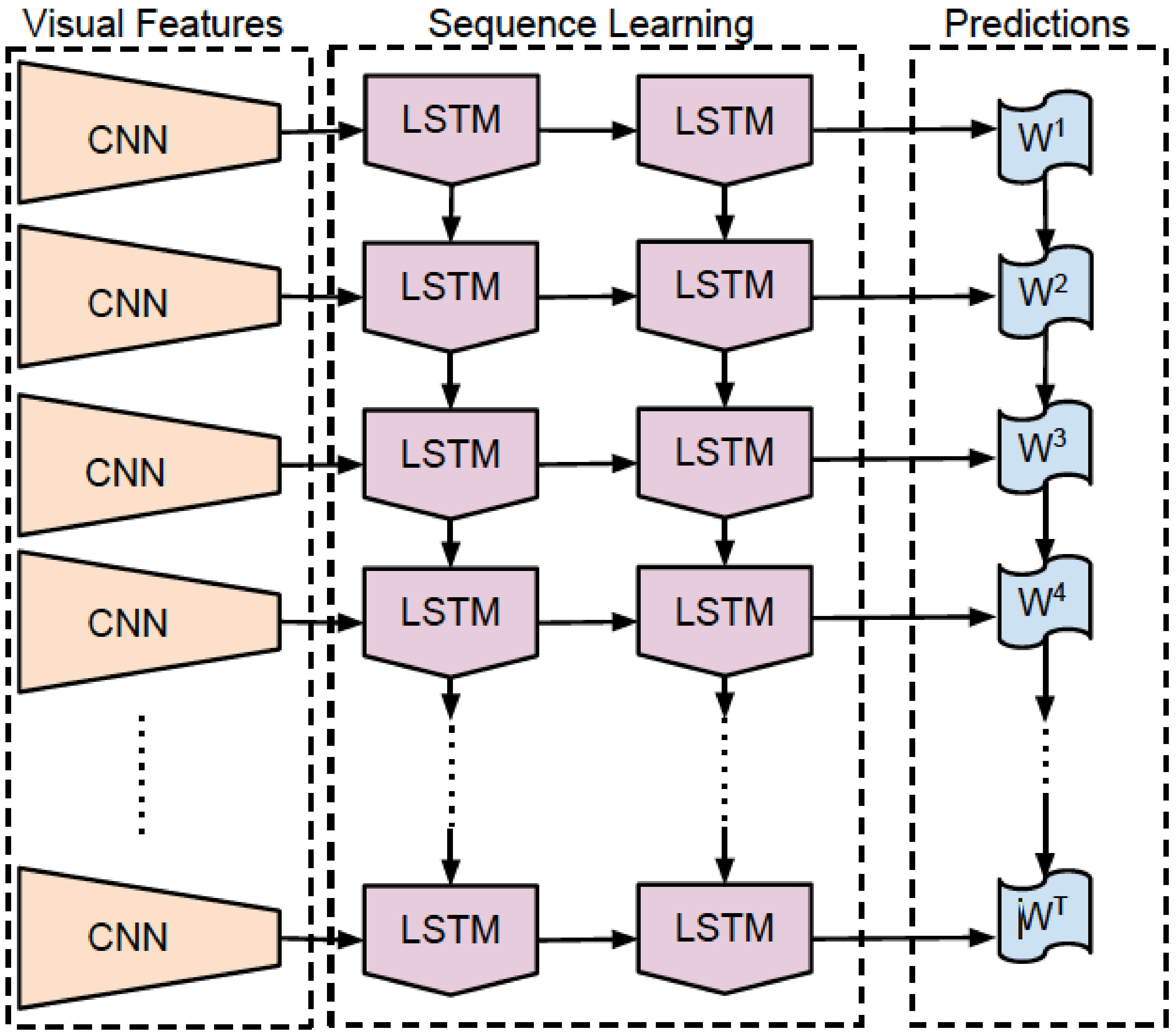
A mostly complete chart of
Neural Networks
©2016 Fjodor van Veen - asimovinstitute.org
Backfed Input Cell
Input Cell
Noisy Input Cell
Hidden Cell
Probablistic Hidden Cell
Spiking Hidden Cell
Output Cell
Match Input Output Cell
Recurrent Cell
Memory Cell
Different Memory Cell
Kernel
Convolution or Pool
Perceptron (P)
Feed Forward (FF)
Radial Basis Network (RBF)
Deep Feed Forward (DFF)
Recurrent Neural Network (RNN)
Long / Short Term Memory (LSTM)
Gated Recurrent Unit (GRU)
Auto Encoder (AE)
Variational AE (VAE)
Denoising AE (DAE)
Sparse AE (SAE)
Markov Chain (MC)
Hopfield Network (HN)
Boltzmann Machine (BM)
Restricted BM (RBM)
Deep Belief Network (DBN)
Deep Convolutional Network (DCN)
Deconvolutional Network (DN)
Deep Convolutional Inverse Graphics Network (DCIGN)
Generative Adversarial Network (GAN)
Liquid State Machine (LSM)
Extreme Learning Machine (ELM)
Echo State Network (ESN)
Deep Residual Network (DRN)
Kohonen Network (KN)
Support Vector Machine (SVM)
Neural Turing Machine (NTM)

Some More Types of Deep Neural Networks

- RBM: Restricted Boltzman Machines (RBM) - older directed deep model.
- RNN: Recurrent Neural Networks (RNN) - allow links from outputs back to inputs, over time, good for time series learning
- LSTM: Long-Term Short-Term networks - more complex form of RNN
  - integrate strategically remembered particular information from the past
  - formalizes a process for forgetting information over time.
  - useful if you need to learn patterns over time and your data features
- DeepRL: Deep Reinforcement Learning
- GAN: General Adversarial Networks - train two networks at once

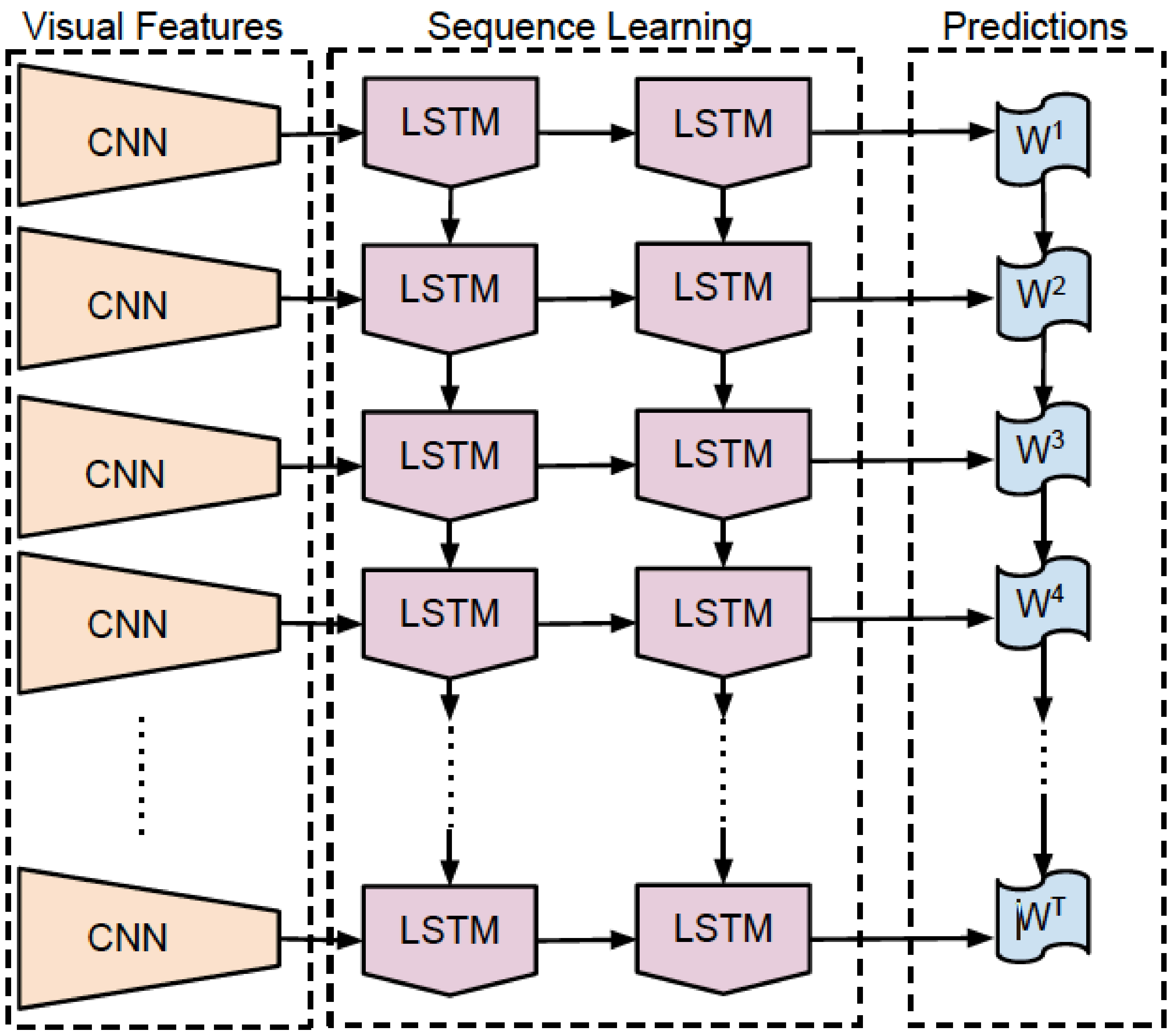
Long-term Recurrent Convolutional Networks (LRCN)

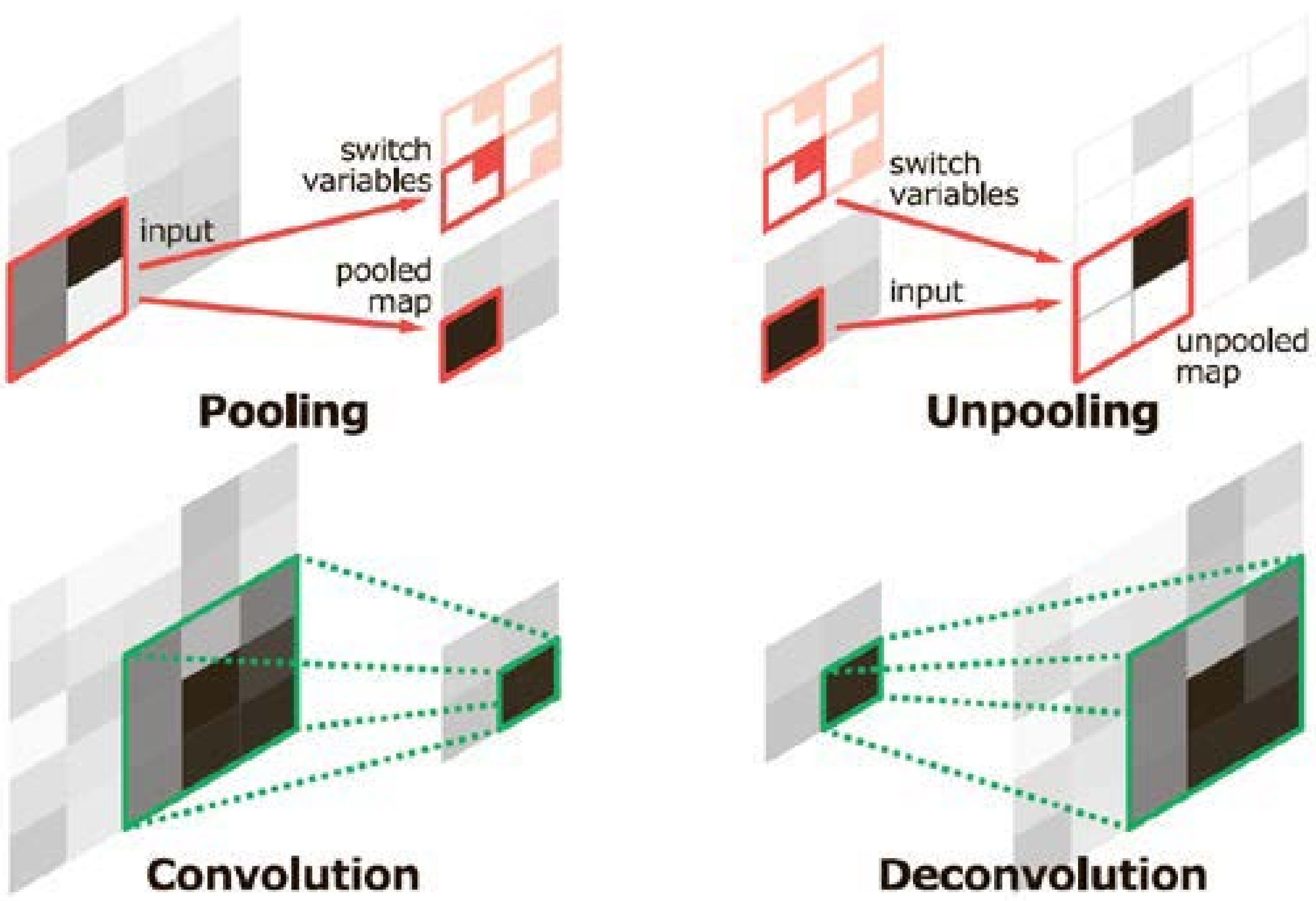
- Combine the abilities of CNNs and LSTMs into an end-to-end architecture
- Applications
  - **Video prediction** - Can learn a generative model which can directly generate a new instance of a later timestep
  - **Activity Recognition -** Assign a class label for a video



Method II

- Treat satellite images as videos, use LRCN to learn generative model for prediction.
- Novel idea: only need to get the spreading process right, not the entire image.





Outline

- My Interests
- Artificial Intelligence, Machine Learning, Big Data and all that...
- Decision Making / Reinforcement Learning
  - Sustainable Harvests, Let Burn Analysis, Fire Treatment Planning
- Deep Learning and Deep RL
  - Learning Spatial Dynamics for forest wildfire
    - Method I : Deep RL
    - Method II : LRCNs
- Future Challenges and Opportunities

Ideas For Future

- Getting more AI/ML people into Fire Management
  - Release datasets? Kaggle competitions?
    - Montesinho Park Fire: https://www.kaggle.com/c/hw2-forest-fires/leaderboard
  - Agent based models repositories
- Inverse Reinforcement Learning (apprenticeship learning)
  - Apply Mikael's idea to policies
  - Give best practices of fire treatments etc
  - Infer the objectives/rewards that would create this policy

References

- McGregor, R Houtman, C Montgomery, R Metoyer, TG Dietterich. "*Fast Optimization of Wildfire Suppression Policies with SMACS*". Arxiv 1703.09391. 2017.
- Christopher J. Lauer, Claire A. Montgomery, Thomas G. Dietterich. "*Spatial interactions and optimal forest management on a fire-threatened landscape*" In Forest Policy and Economics, Volume 83, Pages 107-120. 2017.
- Crowley. "Using Equilibrium Policy Gradients for Spatiotemporal Planning in Forest Ecosystem Management" IEEE Transactions on Computers: Special Edition on Computational Sustainability, 2013.
- Houtman, Montgomery, Gagnon, Calkin, Dietterich, McGreggor and Crowley. "Allowing a wildfire to burn: Estimating the effect on future fire suppression costs". International Journal of Wildland Fire, 2013.
- K. Hall, M. Alkaee Taleghan, H. J. Albers, T. Dietterich, and M. Crowley, "Managing Invasive Species in a River Network," in Third International Conference on Computational Sustainability, 2012.

References

- Crowley and Poole. Policy Gradient Planning for Environmental Decision Making with Existing Simulators. AAAI 2011.
- V. Mnih, A. Puigdomènech Badia, M. Mirza, A. Graves, T. Harley, T. P. Lillicrap, D. Silver, K. Kavukcuoglu, K. Com, and G. Deepmind, “Asynchronous Methods for Deep Reinforcement Learning.” 2016.
- V. Mnih, K. Kavukcuoglu, D. Silver, A. a Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, “Human-level control through deep reinforcement learning,” Nature, vol. 518, no. 7540, pp. 529–533, 2015.

Thank You

Questions?

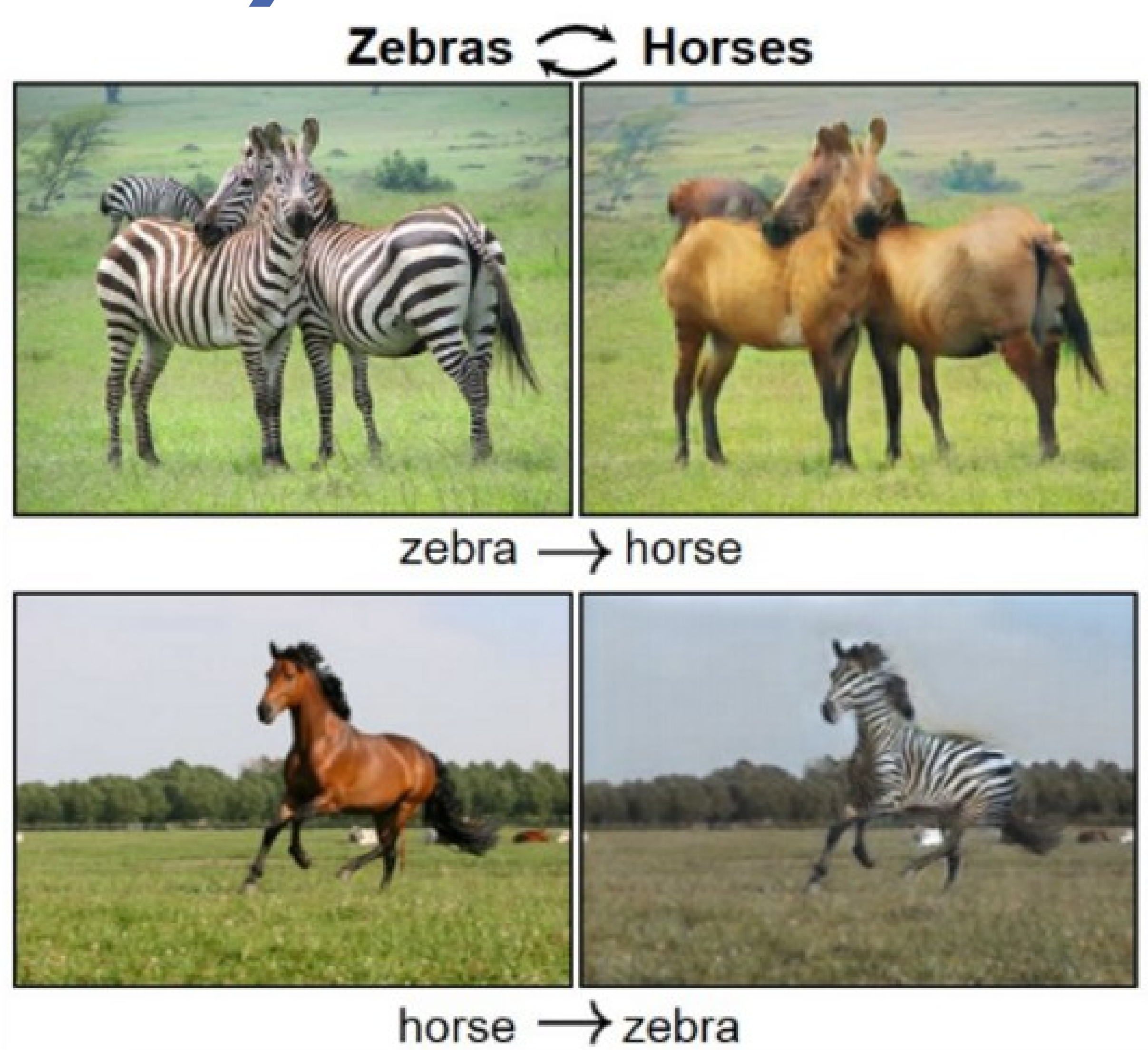
- Students
  - Subramanian Sriram
  - Pardis Zohouri

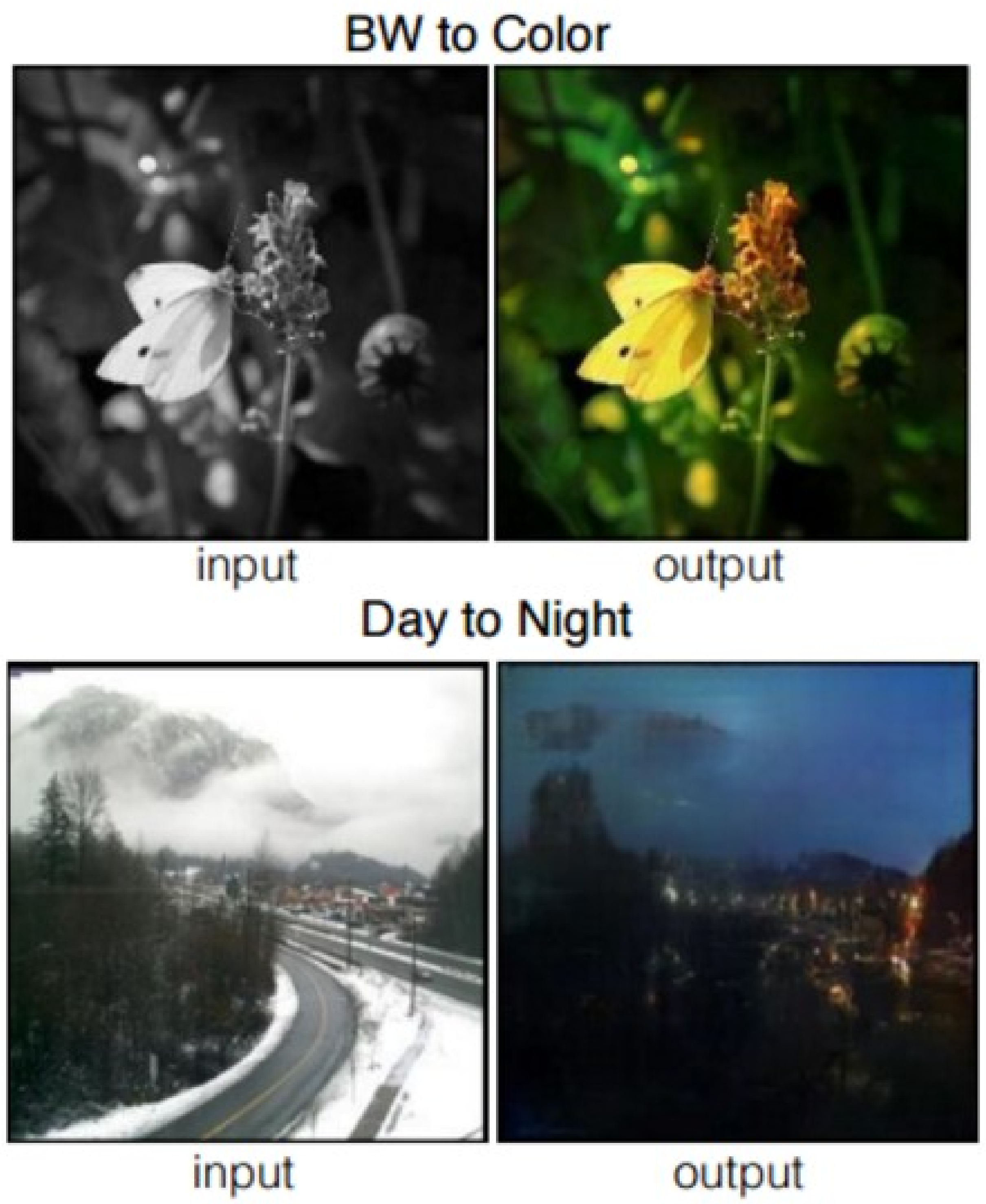
WATERLOO INSTITUTE for COMPLEXITY & INNOVATION

UNIVERSITY OF WATERLOO

NSERC CRSNG

Other Algorithms : **General Adversarial Network (GAN)**





- One network produces/hallucinates new answers (generative)
- Second network distinguishes between the real and the generated answers (adversary/critic)
- How can this approach be used for spatial data beyond photographs?

Future Challenges and Opportunities

Integrating Time, Space and Decisions

- LRCN + Local Policy Representation + Deep RL
- Define a local policy inside the CNN architecture
  - Actions as additional inputs for each pixel/location
- Train the parameters for local policy as part of entire network training via backpropagation

A Short History

- 40's Early work in NN goes back to the 40s with a simple model of the neuron by McCulloh and Pitt as a summing and thresholding devices.
- 1958 Rosenblatt in 1958 introduced the Perceptron,a two layer network (one input layer and one output node with a bias in addition to the input features.
- 1969 Marvin Minsky: 1969. Perceptrons are 'just' linear, AI goes logical, beginning of "AI Winter"
- 1980s Neural Network resurgence: Backpropagation (updating weights by gradient descent)
- 1990s SVMs! Kernals can do anything! (no, they can't)

A Short History of Neural Networks

- 1993 LeNet 1 for digit recognition
- 2003 Deep Learning (Convolutional Nets Dropout/RBMs, Deep Belief Networks)
- 1986, 2006 Restricted Boltzman Machines
- 2006 Neural Network outperform RBF SVM on MNIST handwriting dataset (Hinton et al.)
- 2012 AlexNet for ImageNet challenge - this algorithm beat competition by error rate of 16% vs 26% for next best
  - ImageNet : contains 15 million annotated images in over 22,000 categories.
  - ZFNet paper (2013) extends this and has good description of network structure
- 2012-present Google Cat Youtube, speech recognition, self driving cars, computer defeats regional Go champion, ...
- 2014 GoogLeNet added many layers and introduced inception modules (allows parallel computation rather than serially

A Short History of Neural Networks

- 2014 Generative Adversarial Networks (GANs) introduced.
- 2015 Microsoft algorithm beats human performance at ImageNet challenge.
- 2016 AlphaGo defeats one of best world players of Go Lee Sedol using Deep Reinforcement Learning.
- 2016 Deep Mind introduces A3C Deep RL algorithm that can learn to play Atari games from images by playing with no instructions.

Heuristics for Improving Training

There are a number of useful but not necessarily theoretically justified tricks (ie. heuristics) for training Neural Networks that are useful in practice.

- Less hidden nodes, just enough complexity to work, not too much to overfit.
- Train multiple networks with different sizes and search for the best design.
- Validation set: train on training set until error on validation set starts to rise, then evaluate on evaluation set.
- Try different activation functions: sigmoid or ReLU Randomly choose subsets of the data to training.
- Dropout (Hinton 2014) - randomly ignore certain units during training, don't update them via gradient descent, leads to hidden units that specialize
- Modify learning rate over time (cooling schedule)

PAC MDPs for Invasive Species Control

Tamarisk plants invading river networks.

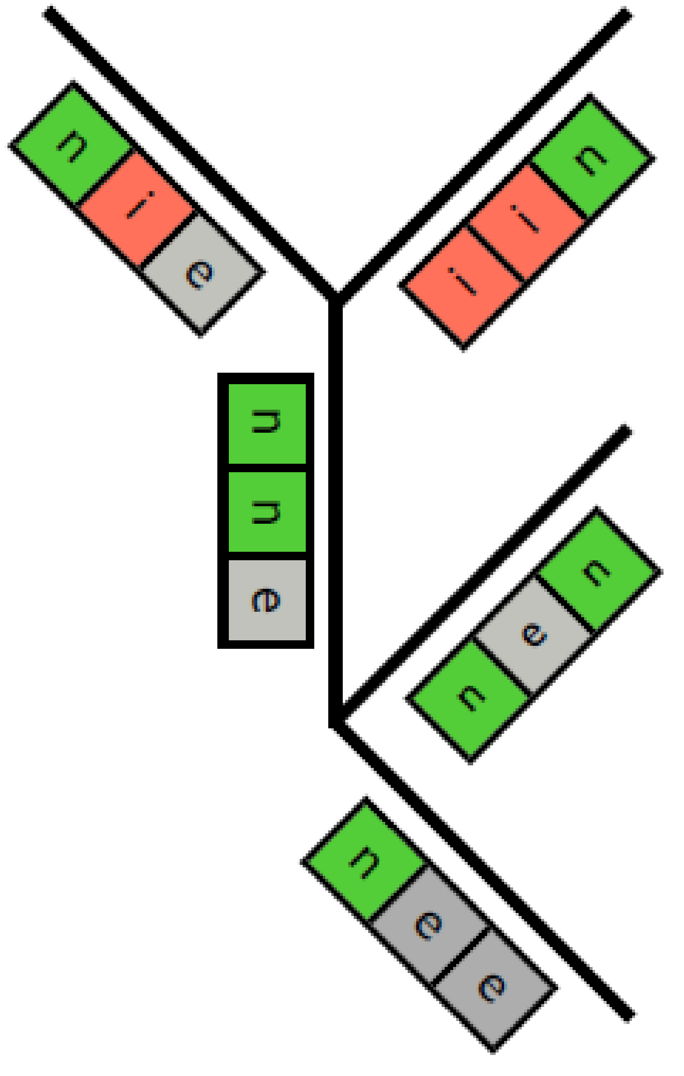
Reaches: E reaches with H sites each
E ranges in [1,7], H in [1,5]

Site States: M={native, invading, empty}

Total States: ~3^{EH}

Actions:A={plant native, eradicate invaders, do both, do nothing}
Total Actions :4^{E}

Dynamics: Provided by stochastic local spreading formulas for each 'propagule'.
Simulator was part of 2013 RL Competition



What if you want the optimal solution? Or hard guarantees?