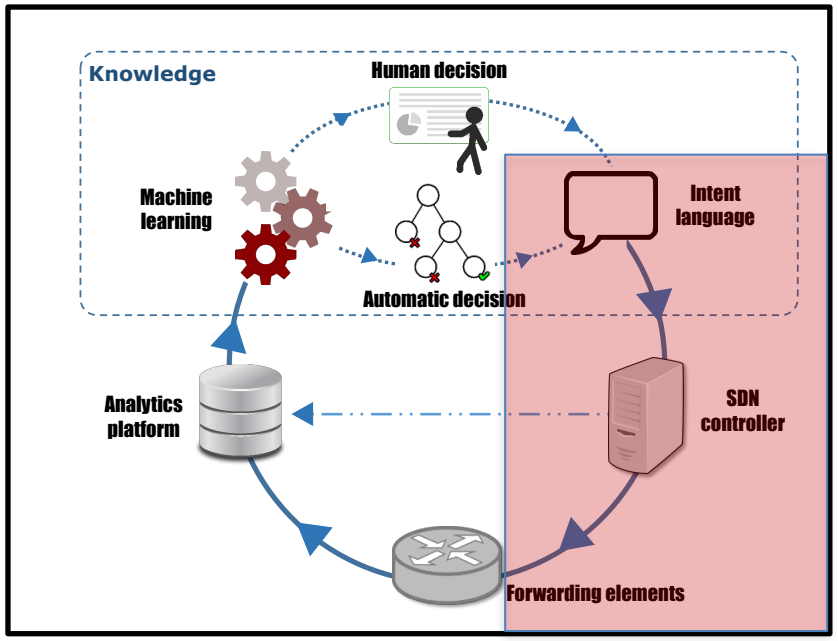
# So What Do We Need To Do?

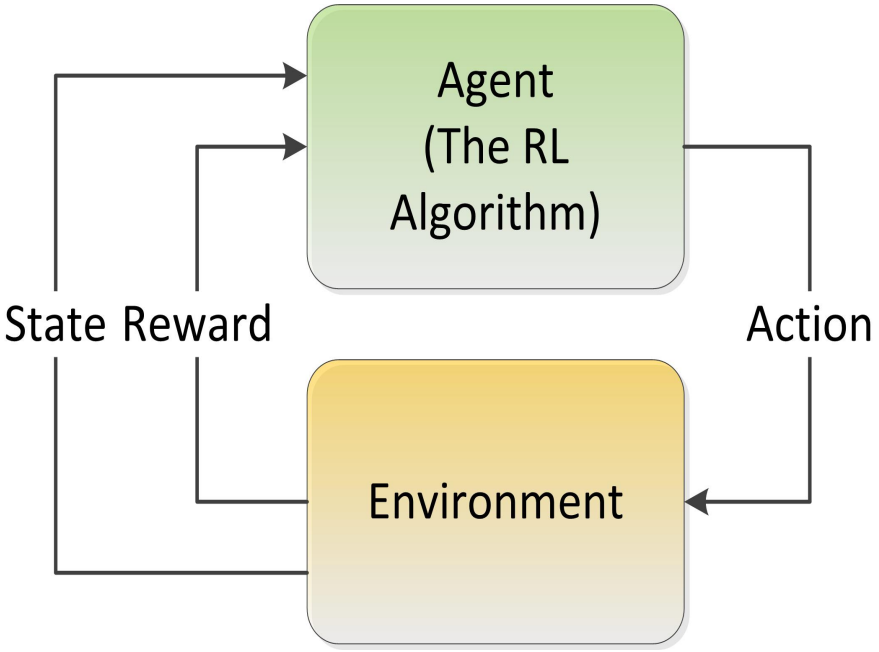
- Find a Useful Theory of Networking
  - As opposed to ad-hoc approaches
  - Again, consider the success of convolutional neural networks
  - What is the “interlingua” for networking?
- Public Data Sets
- Skill Sets
  - But watch out
  - https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf
- Apply state of the art machine learning algorithms
- Build ML platforms and associated automation
- Explainable end-to-end systems
- Learn control not just prediction

# Learning Control?

***Reinforcement Learning Meets Monte Carlo Tree Search and Deep Learning***



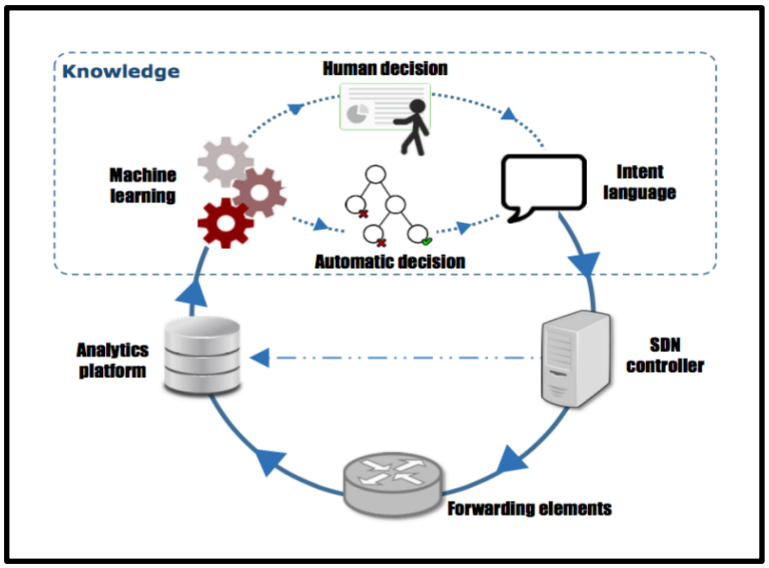

Hard coded control logic (not learned)



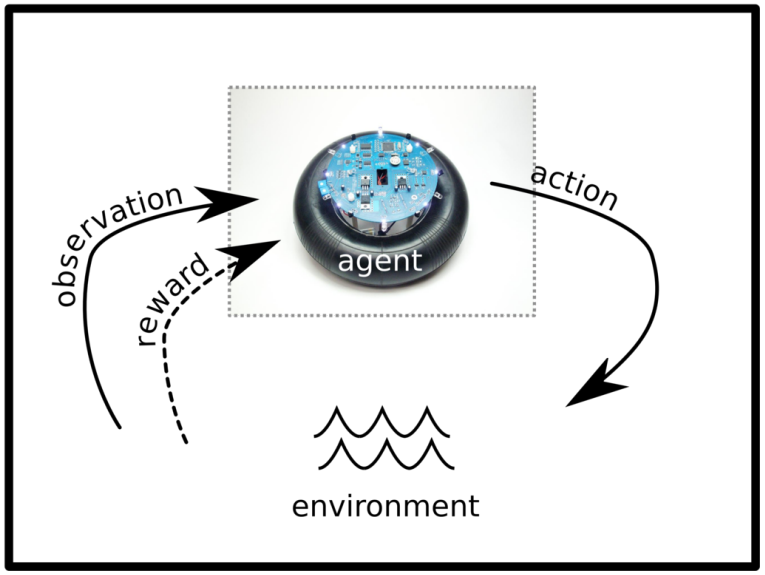

- Security: Agent can learn dynamic/evolving behavior of adversary
- DevOPs: Agent can learn workflow automation (e.g. Openstack Mistral)
- Orchestration: Agent can learn dynamic behavior of VNFs/system
- Deep policy net, $\pi(s) = a, s \in S, a \in A(s)$, can capture human intuition

http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html

# Static vs. Reinforcement Learning Architectures




- Today's Static ML Architectures
  - ML doesn't have "agency"
  - Hard-coded or open loop control
- Learns prediction
- Doesn't learn control
- Assumes stationary DGD
- Largely off-line




- Reinforcement Learning Architecture
  - Agent architecture
  - Agent learns control/action selection
- Adapts to evolving environment
- Non-stationary distributions
- *Network gamification*

# Summary

- Lots of Network Data
  - Standardized and labeled datasets still scarce
  - However, most network data sources (e.g., netflow) not designed for ML
  - Arbitrary data might be over-valued

- Lots of open source ML frameworks, Cloud APIs, Examples, …
  - Google -- Tensorflow (tensorflow.org)
  - Facebook -- Torch (torch.ch)
  - Microsoft -- CNTK (https://github.com/Microsoft/CNTK)
  - Amazon -- MXNET (https://aws.amazon.com/mxnet)
  - Other popular frameworks
    - Keras (https://keras.io/)
    - Theano (https://github.com/Theano)
    - …

- However, networking industry still dominated by MLwashing, platforms, clustering, …

- Skills gap persists and it still requires skill/experience to
  - Build DNN architectures/models
  - Find ”good” settings for hyper-parameters
  - Prevent overfitting
  - But again, be careful: SOTA LSTMs+RLs are starting to be able to build neural networks
    - https://openreview.net/pdf?id=r1Ue8Hcxg, http://papers.nips.cc/paper/6461-learning-to-learn-by-gradient-descent-by-gradient-descent.pdf, …
  - …

- All of this said, you *will* be seeing ML in all facets of networking
  - And everything else for that matter

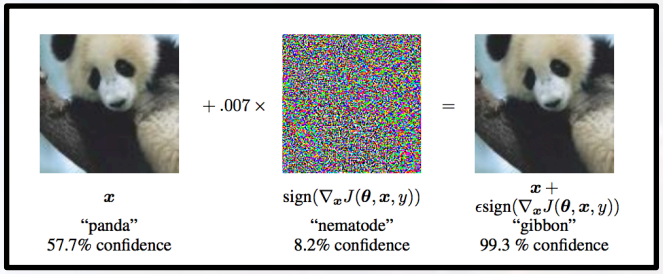
# So Call To Action

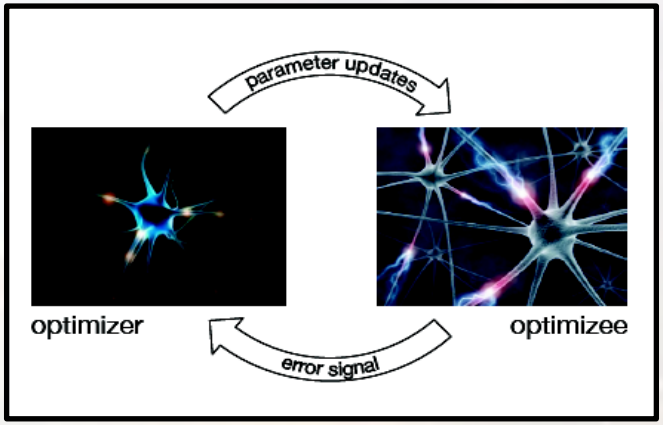
## Where Do We Need To Focus?

- Find a Useful Theory of Networking
  - As opposed to ad-hoc approaches
  - Again, consider the success of convolutional neural networks
  - What is the “interlingua” for networking?
- Public Data Sets
- Skill Sets
- State of the art machine learning algorithms
- ML platforms and associated automation
- Explainable end-to-end systems
- Learn control not just prediction
  - Reinforcement Learning
  - Evolution Strategies (https://blog.openai.com/evolution-strategies)
  - Others?

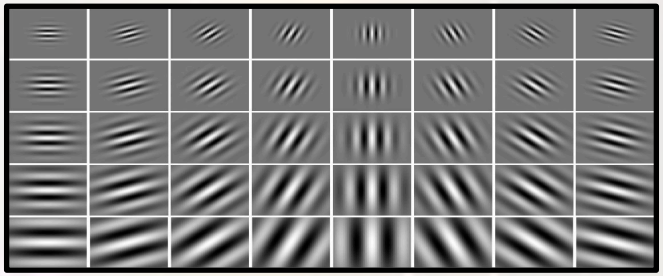
# Finally…Many Beautiful Mysteries Remain

- Back-propagation
  - Gradient-based optimization/back-propagation
  - Why is such a simple algorithm so powerful?
  - New optimizer/optimizee paradigms
    - AlphaGo
      - https://deepmind.com/research/alphago
    - Adversarial Images/Generative Adversarial Nets (GANs)
      - https://arxiv.org/abs/1412.6572
    - Learning to learn by gradient descent by gradient descent
      - https://arxiv.org/abs/1606.04474
    - Neural Architecture Search With Reinforcement Learning
      - https://openreview.net/pdf?id=r1Ue8Hcxg




- Neural Nets
  - What are the units (artificial neurons) actually doing?
  - https://arxiv.org/pdf/1509.06321.pdf



- Why does deep and cheap learning work so well?
  - Physics perspective
  - http://arxiv.org/pdf/1608.08225v1.pdf

- So how many 28x28 grayscale images are there?
- Say there are 8 bits of grayscale → $256^{784}$ possible images
- Our autoencoder had 156800 parameters and $156800/256^{784} \approx 0$
- So how can a simple autoencoder find the MNIST subspace?