# Learning Generalized Heuristics Using Deep Neural Networks

Julia Nakhleh
Director: Siddharth Srivastava

AAIR
Autonomous Agents and Intelligent Robots
Arizona State University

# Introduction

- Classical planning
  - AI domain concerned with decision-making in real-world tasks (e.g. doing laundry)
  - Computationally difficult due to size and complexity of environments
    - Slight increase in number of objects → significant slowdown
  - PSPACE-complete even when environment is deterministic and fully-observable (Bylander 1994)
- Generalized planning
  - Construction of plans that solve multiple problem instances
  - Identify common structures between problems; execute same solution with minor modifications
  - Improves efficiency on new planning instances

# Introduction (cont.)

- Our approach: generalized planning + deep learning
    - Given a planning problem instance and a generalized representation of a state, our network learns a Generalized Reactive Policy (GRP) that predicts the best action to take from that state
    - Network also predicts the generalized *role* of the object within the environment
- Tested on two domains: blocks world and logistics
    - Results in both domains indicate success, although network accuracy is lower in the more complex logistic domain

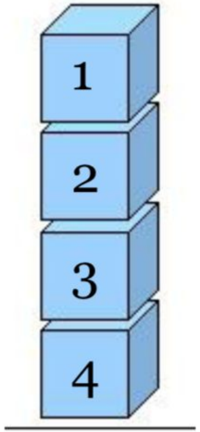
# Related Work

- Application of learning techniques to classical planning interfaces
  - Martin and Geffner (2004): learn generalized policies using concept languages
  - Rosman and Ramamoorthy (2012): use reinforcement learning to learn action priors
- Deep Neural Networks
  - State-of-the-art in image classification (Krizhevsky, Sutskever, and Hinton 2012) and NLP (Sutskever, Vinyals, and Le 2014)
  - Learn heuristic functions (Ernandes and Gori 2004)
  - Learn an Alpha Go policy (Silver et al. 2017)
  - Learn a GRP and a planning heuristic based on images of a successful Sokoban trace (Groshev et al. 2017)

# State Abstraction

- Given a domain, a goal formula, and a set of initial states, a *generalized planner* computes a generalized plan that solves all of the initial states
    - Initial states may be from different state spaces, but must be from the same domain
- Srivastava, Immerman and Zilberstein (2011) propose a method of state abstraction that compactly represents concrete, unbound states
    - Summary elements: may represent multiple physical entities
    - Non-summary elements: represent only one physical entity

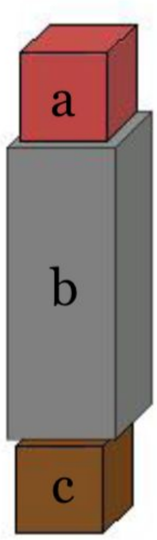
# Non-abstract state




| on | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 0 | 1 |
| 4 | 0 | 0 | 0 | 0 |

Non-abstracted representation of a blocks world state (Srivastava 2011).

# Abstract state

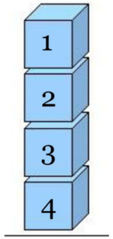



| *on* | a | b | c |
|---|---|---|---|
| a | 0 | ½ | 0 |
| b | 0 | ½ | ½ |
| c | 0 | 0 | 0 |

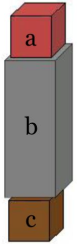
Abstract representation of a blocks world state (Srivastava 2011).

# State Abstraction (cont.)

- Three-valued logic
  - 1 = true, 0 = false, ½ = unknown/undefined
- Role = the set of unary predicates an object satisfies
  - Roles correspond to summary elements
  - a = {*clear*}, c = {*onTable*}, b = {} (i.e. ¬*clear* ^ ¬*onTable*)
  - Summary element *a* corresponds to block 1, *c* to block 4, and *b* to blocks 2 and 3



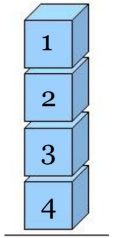

| *on* | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 0 | 1 |
| 4 | 0 | 0 | 0 | 0 |



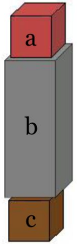

| *on* | a | b | c |
|---|---|---|---|
| a | 0 | ½ | 0 |
| b | 0 | ½ | ½ |
| c | 0 | 0 | 0 |

# State Abstraction (cont.)

- Binary relationships become imprecise
  - For example, (on a b) is valued as ½ because not all of the concrete elements within a are directly on top of all of the concrete elements within b




| *on* | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| **1** | 0 | 1 | 0 | 0 |
| **2** | 0 | 0 | 1 | 0 |
| **3** | 0 | 0 | 0 | 1 |
| **4** | 0 | 0 | 0 | 0 |




| *on* | a | b | c |
|---|---|---|---|
| **a** | 0 | ½ | 0 |
| **b** | 0 | ½ | ½ |
| **c** | 0 | 0 | 0 |

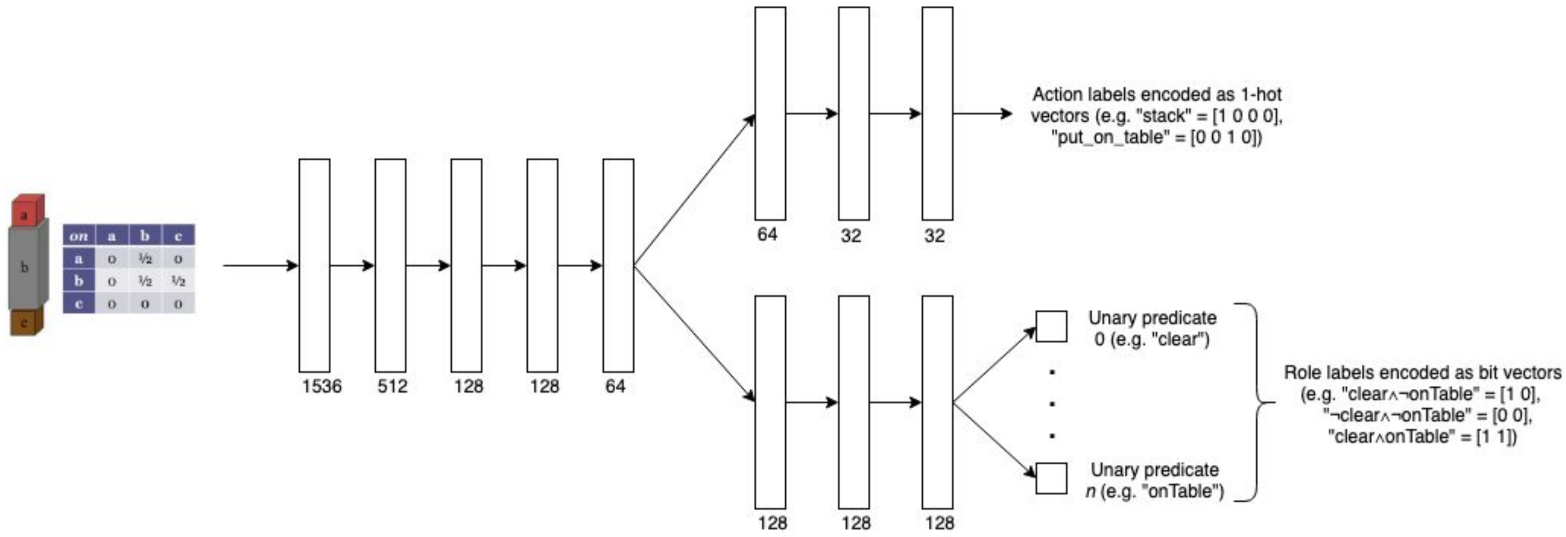
# Neural Network Input/Output

- Input format: matrix representing abstracted state
  - Dimensions: $n_r$ x $n_r$ x $n_b$
  - $n_r$ = number of possible roles, $n_b$ = number of binary predicates in domain
  - We used abstracted states as input because they allow an unbounded number of objects to be represented compactly
    - Problems of different sizes are represented with same dimensions (necessary for NN)
- Label (i.e. output) format: vectors representing (i) correct action to take from a given state and (ii) role of object being acted upon
  - Actions: 1-hot encoded vector of length $n_a$ (number of actions in domain)
  - Roles: bit-encoded vector of length $n_u$ (number of unary predicates in domain)

# Neural Network Design

- 11 + *n* fully-connected square layers, where *n* is the number of unary predicates in the domain
  - 5 shared layers
  - 3 action layers
  - 3 roles layers
  - *n* 1-bit layers, each of which indicates whether a unary predicate is present (1) or absent (0) in the role
- Loss taken as softmax over loss on each unary predicate layer + loss on action predictions

# Neural Network Design (cont.)



# Domains Tested

- Blocks world
  - Robot must stack/unstack a set of wooden blocks on a table
  - Only one block can be moved on a time
  - Start: 2-10 blocks in random configuration of ≥1 towers
  - Goal: all blocks on table
- Logistic
  - Truck(s) must load, unload, and deliver crates between a set of fixed locations
    - Our version uses 1 truck, 5 locations, and 15-20 crates
    - Truck assumed to have infinite capacity
  - Start: all crates are randomly distributed among locations
  - Goal: crates are evenly distributed among locations

# Experiments

- Fast-Forward (FF) Planner (Hoffman and Nebel 2001)
  - Given initial configuration, goal configuration, and the rules of the planning domain, FF computes sequence of actions and intermediate states from start to goal
  - [state i: “unstack 1”] → [state ii: “place_on_table 4”] → …
  - NN data (abstract states) and labels (action and concrete object role) are manually extracted from returned FF Plan
- NN training data generated by running FF on large problem sets
  - Blocks world: 6000 training problems, 1000 test problems
    - ~35,000 training instances, ~5,762 test instances
  - Logistic:  500 training problems, 100 test problems
    - ~30,000 training instances, ~6,000 test instances

# Results - Blocks World

- Proof of concept
- Immediately achieves 100% accuracy on both action and role predictions
- All loss in range [-2e-07, 3e-06] → effectively ~0.00 loss
- Extremely high accuracy due to simplicity of problem
    - In our version, only 2-10 blocks
    - Since goal is “all blocks on table”, the network only ever learns action “unstack” and role “clear”
    - Still a promising result

# Results - Logistics

- Accuracy of action predictions reaches 96%
- Accuracy of role predictions hovers around 75%
  - Partially due to the way that current locations of crates within trucks are encoded within the domain

# Logistics - sample predictions


Correct action: unloadAtL5 crate5
Network prediction: unloadAtL5 crate5

Role of crate 5: ["crate", "atL3", "destinationL5"]
Network prediction: ["crate"]

# Logistics - sample predictions


Correct action: unloadAtL5 crate10
Network prediction: unloadAtL5 crate10

Role of crate 10: ["crate", "atL3", "destinationL5"]
Network prediction: ["crate"]

# Logistics - sample predictions


Correct action: unloadAtL4 crate14
Network prediction: unloadAtL4 crate14

Role of crate 14: ["crate", "atL1", "destinationL4"]
Network prediction: ["crate"]

# Logistics - sample predictions


Correct action: moveToL5 truck1
Network prediction: moveToL5 truck1

Role of truck1: ["truck", "atL4"]
Network prediction: ["truck", "atL4"]

# Future Work

- Experiment with different loss functions (sum of squares, etc.)
  - Weight action loss vs. role loss
- Test on different domains and problem types
  - Blocks world: different goal configurations, block colors
  - Logistics: multiple trucks, limited truck capacity
- Convolutional Neural Network (CNN)
  - Particularly useful if input is represented as an image rather than a matrix
- Test effectiveness of learned GRP as a planning heuristic
  - Compare performance against best-first search

# Conclusion

- Neural network indicates a promising ability to learn based on abstract state representations
    - More difficult in more complex domains
    - We hypothesize that with further tuning, and with slight modifications to the logistics domain, the network's accuracy on the logistics domain can be significantly improved
- Further testing is needed to compare effectiveness of learned GRP as a planning heuristic

# Acknowledgements

Thank you to Dr. Srivastava for allowing me to work with the AAIR lab this year and for serving as my director on this project.

Thanks as well to all the members of the lab, particularly Naman Shah and Mathew Morales, who contributed significantly to this research.

And thank you Dr. Fainekos for serving on my committee!

# References


Abel, D., Hershkowitz, D. E., Barth-Maron, G., Brawner, S., O'Farrell, K., MacGlashan, J., & Tellex, S. (2015). Goal-based Action Priors. In *Proceedings of the Twenty-Fifth International Conference on International Conference on Automated Planning and Scheduling* (pp. 306–314). Jerusalem, Israel: AAAI Press. Retrieved from http://dl.acm.org/citation.cfm?id=3038662.3038705

Bylander, T. (1994). The computational complexity of propositional STRIPS planning. *Artificial Intelligence*, *69*(1), 165–204. https://doi.org/10.1016/0004-3702(94)90081-7

Duan, Y., Andrychowicz, M., Stadie, B. C., Ho, J., Schneider, J., Sutskever, I., … Zaremba, W. (2017). One-Shot Imitation Learning. *ArXiv:1703.07326 [Cs]*. Retrieved from http://arxiv.org/abs/1703.07326

Ernandes, M., & Gori, M. (2004). Likely-admissible and Sub-symbolic Heuristics. In *Proceedings of the 16th European Conference on Artificial Intelligence* (pp. 613–617). Amsterdam, The Netherlands, The Netherlands: IOS Press. Retrieved from http://dl.acm.org/citation.cfm?id=3000001.3000130

Groshev, E., Goldstein, M., Tamar, A., Srivastava, S., & Abbeel, P. (2017). Learning Generalized Reactive Policies using Deep Neural Networks, 12.

Hoffmann, J., & Nebel, B. (2001). The FF Planning System: Fast Plan Generation Through Heuristic Search. *Journal of Artificial Intelligence Research*, *14*, 253–302. https://doi.org/10.1613/jair.855

Hu, Y., & De Giacomo, G. (2011). Generalized Planning: Synthesizing Plans that Work for Multiple Environments. *IJCAI*, 6.

Konidaris, G. (2006). A Framework for Transfer in Reinforcement Learning.

Konidaris, G., Scheidwasser, I., & Barto, A. G. (2012). Transfer in Reinforcement Learning via Shared Features. *J. Mach. Learn. Res.*, *13*, 1333–1371.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1* (pp. 1097–1105). USA: Curran Associates Inc. Retrieved from http://dl.acm.org/citation.cfm?id=2999134.2999257

Martín, M., & Geffner, H. (2004). Learning Generalized Policies from Planning Examples Using Concept Languages. *Applied Intelligence*, *20*(1), 9–19. https://doi.org/10.1023/B:APIN.0000011138.20292.dd

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., … Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533. https://doi.org/10.1038/nature14236

Mülling, K., Kober, J., Kroemer, O., & Peters, J. (2013). Learning to select and generalize striking movements in robot table tennis. *The International Journal of Robotics Research*, *32*(3), 263–279. https://doi.org/10.1177/0278364912472380

Pomerleau, D. A. (1988). ALVINN: An Autonomous Land Vehicle in a Neural Network. In *Proceedings of the 1st International Conference on Neural Information Processing Systems* (pp. 305–313). Cambridge, MA, USA: MIT Press. Retrieved from http://dl.acm.org/citation.cfm?id=2969735.2969771

Rosman, B., & Ramamoorthy, S. (2012). What good are actions? Accelerating learning using learned action priors. *2012 IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, 1–6. https://doi.org/10.1109/DevLrn.2012.6400810

Ross, S., Gordon, G. J., & Bagnell, J. A. (2011). A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. *AISTATS*. Retrieved from http://arxiv.org/abs/1011.0686

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., … Hassabis, D. (2017). Mastering the game of Go without human knowledge. *Nature*, *550*(7676), 354–359. https://doi.org/10.1038/nature24270

Srivastava, S. (2011). *Hybrid Search for Generalized Plans Using Classical Planners*. University of Massachusetts, Amherst.

Srivastava, S., Immerman, N., & Zilberstein, S. (2011). A new representation and associated algorithms for generalized planning. *Artificial Intelligence*, *175*(2), 615–647. https://doi.org/10.1016/j.artint.2010.10.006

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2* (pp. 3104–3112). Cambridge, MA, USA: MIT Press. Retrieved from http://dl.acm.org/citation.cfm?id=2969033.2969173

Tamar, A., Wu, Y., Thomas, G., Levine, S., & Abbeel, P. (2017). Value Iteration Networks. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence* (pp. 4949–4953). Melbourne, Australia: International Joint Conferences on Artificial Intelligence Organization. https://doi.org/10.24963/ijcai.2017/700

Tesauro, G. (1995). Temporal Difference Learning and TD-Gammon. *Commun. ACM*, *38*(3), 58–68. https://doi.org/10.1145/203330.203343

Weber, T., Racanière, S., Reichert, D. P., Buesing, L., Guez, A., Rezende, D. J., … Wierstra, D. (2017). Imagination-Augmented Agents for Deep Reinforcement Learning. *ArXiv:1707.06203 [Cs, Stat]*. Retrieved from http://arxiv.org/abs/1707.06203

Yoon, S., Fern, A., & Givan, R. (2002). Inductive Policy Selection for First-order MDPs. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence* (pp. 568–576). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. Retrieved from http://dl.acm.org/citation.cfm?id=2073876.2073944