Decomposing the Prediction Problem; Autonomous Navigation by neoRL Agents

Per Roald Leikanger

UiT – Norges Artigste Universtet
Per.Leikanger@uit.no

Abstract

Navigating the world is a fundamental ability for any living entity. Accomplishing the same degree of freedom in technology has proven to be difficult. The brain is the only known mechanism capable of voluntary navigation, making neuroscience our best source of inspiration toward autonomy. Assuming that state representation is key, we explore the difference in how the brain and the machine represent the navigational state. Where Reinforcement Learning (RL) requires a monolithic state representation in accordance with the Markov property, Neural Representation of Euclidean Space (NRES) reflects navigational state via distributed activation patterns. We show how NRES-Oriented RL (neoRL) agents are possible before verifying our theoretical findings by experiments. Ultimately, neoRL agents are capable of behavior synthesis across state spaces – allowing for decomposition of the problem into smaller spaces, alleviating the curse of dimensionality.

Introduction

Autonomy or any form of self-governed activity implies an ability to adapt with experience; hard-coded algorithms, agents governed by external control, or deterministic model-based path planning can hardly be said to be autonomous. "Navigation can be defined as the ability to plan and execute a goal-directed path" (Solstad, 2009). Robot motion planning can be defined in similar terms (Latombe, 2012); however, cybernetics and robot motion control involves model with limited validity intervals or algorithms for deterministic control. The reward hypothesis from Reinforcement Learning (RL) is relevant in this context: *"That all of what we mean by goals and purposes can be well thought of as maximization of the expected value of the cumulative sum of a received scalar signal (reward)." (Sutton and Barto, 2018).* With a proven track record for learning to solve digital challenges or for intelligent games, RL agents have demonstrated a capability of autonomy for specific challenges. Via methods from function approximation by Deep Learning, methods from RL can form agents with superhuman abilities for certain board games (Tesauro, 1994; Silver et al., 2016, 2017) and games of hazard (Heinrich and Silver, 2016). However, RL supported by deep function approximation is known to require a tremendous amount of training: Robot autonomy by RL remains an unsolved challenge, partially due to requirements for real-time execution and model-uncertainty – limiting the number of accurate samples for training (Kober et al., 2013). RL agents supported by deep function approximation can learn impressive abilities, but statistical machine learning approaches require much experience, do not generalize well, and are monolithic during training and execution (Kaelbling, 2020).

Autonomous navigation is an ability unique to the central nervous systems in the animal and insects. Determining one's parameter configuration relative to an external reference, one's *allocentric* coordinate, is critical for navigation learning (Whitlock et al., 2008). Several mechanisms have been identified in the brain that represent Euclidean coordinates at the single-neuron level (Bicanski and Burgess, 2020). Notable examples for navigation are Object Vector Cells (Høydal, 2020), representing the allocentric location of objects around the animal, Head-Direction Cells (Taube et al., 1990), representing the heading of the animal, and border cells (Solstad, 2009), representing the proximity of borders for navigation. Possibly the most well-known cell for Neural Representation of Euclidean Space (NRES) is the *Place Cell*. This first identified NRES modality represents the allocentric location of the animal (O'Keefe and Dostrovsky, 1971): When an animal's location is within the *receptive field* of one place cell, the neuron is active in terms of having a heightened firing frequiency. The activation pattern in an appropriate population of NRES neurons can thus map any position in a finite Euclidean space (Fyhn et al., 2004). Other NRES modalities have later been identified, with a similar mechanism for representing coordinates in other Euclidean spaces (Bicanski and Burgess, 2020). With our sense of orientation originating from multiple NRES modalities, distributed representation of state appears to be of critical importance for navigational autonomy.

This article starts out by presenting important considerations from RL and directions that could allow for a distributed representation of state. Off-policy learning allows

agents to learn general value functions for independent aspects of a task (Sutton et al., 2011). When a hoard of learners base their value function on a mutually exclusive reward signal, inspired by NRES cells, we propose a method for learning an orthogonal basis for behavior. Experiments with NRES-Oriented RL (neoRL) agents by the Place Cell NRES modality demonstrate how the proposed framework allows for reactive navigation in real-time.

Interaction learning by RL in AI

Reinforcement learning is the direction in machine learning concerning learning behavior through interaction with an environment. We say that the decision *agent* learns to achieve a task according to a scalar reward signal $\mathbb{R}$ by interaction with an *environment*. The accumulated experience takes the form of agent *value function*, reflecting the benefit of visiting different states or state-actions pairs according to the *reward signal* during training. When the algorithm learns the value of state-action pairs, i.e., learning the value of selecting specific actions from different states, this is referred to as Q-learning. An important aspect of RL environments is the *Markov property*: When a state-action pair uniquely defines the probability distribution of the next state, the decision process is referred to as a Markov Decision Process (MDP). When a problem can be represented as an MDP, an RL-agent can, in theory, learn an optimal solution to tasks expressed by a reward function from interaction alone (Sutton and Barto, 2018).

The *prediction problem* in reinforcement learning concerns estimating the value of visiting different states s while following policy π. The agent state is a compact representation of the history and necessary information for the agent to make a decision at time t. The value function can be updated according to the Bellman equation:

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_\pi(s')\right] \quad (1)$$

Updating the value function under policy π from experience gathered while following the policy π, is referred to as on-policy learning (Sutton and Barto, 2018). Off-policy learning allows an agent to form the value function while following another behavior policy. Through off-policy learning, an agent can learn the value function under a target policy π_t while following a different behavior policy $\pi_b \neq \pi_t$. The agent can, for example, initially follow a more exploratory policy or learn while observing human control (Abbeel et al., 2007). Learning the value function is possible through pure observation.

General Value Function (GVF) is one identified use of off-policy learning, where the agent learns value functions potentially unrelated to the control problem (Sutton et al., 2011). These partial agents, only concerned with accumulating experience, can be seen as independent *learners* of an auxiliary value function used to answer questions about the environment. Examples of questions, as listed in the original paper, could be time-to-obstacle or time-to-stop for the Critterbot demonstration (Sutton et al., 2011). Auxiliary value functions can also be directly involved in policy, as demonstrated for the Atari game Ms. PacMan. A set of General Value Functions were trained for manually designed subchallenges in the Ms. Packman computer game, resulting in an exponential breakdown of problem size compared to "single-headed" RL agents (Van Seijen et al., 2017). Wiering and Van Hasselt (2008) gave a methodological overview over ensemble methods for integrating experience from multiple algorithms when forming policies. Notably, Boltzmann addition and Boltzmann multiplication could integrate policies from multiple sources before action selection (Wiener, 1948). Both Wiering and Van Hasselt (2008) and Van Seijen et al. (2017) propose ways multiple off-policy learners could be involved in forming policy. From these demonstrations on how multi-learner agents are possible, we shall dive further into the mechanism of behavior synthesis. But first, some neuroscience.

Neural Representation of Euclidean Space

The 1906 Nobel price in physiology and medicine was awarded Santiago Ramón Y Cajal for work initiating the neuron doctrine (Ramón y Cajal, 1911), claiming that behavior originates from a network of cells with signaling capabilities rather than a monolithic soul. The neuron doctrine supplied a mechanistic understanding of biological computation as a distributed network of weak computational units. Only by network phenomena and a delicately connected net of neurons can decisions, policies, and ultimately behavior emerge. Eric Kandel later reported how synaptic connections change with use and how learning and memory are consequences of synaptic plasticity (Kandel and Tauc, 1965). Before the neuron doctrine, the consensus was that behavior and decision-making originate from a monolithic entity that followed us in this life and beyond – *the soul*.

Neural Representation of Euclidean Space (NRES) have been reported for different Euclidean spaces on a per-neuron cellular activation: when the Euclidean coordinate falls within the receptive field of an NRES neuron, the neuron fires with a heightened firing frequency. A growing number of NRES modalities have been identified, with notable examples for navigation being place cells (O'Keefe and Dostrovsky, 1971), head-direction cells (Taube et al., 1990), and object-vector cells (Høydal, 2020). While some NRES neurons have simple receptive fields centered around a coordinate, others have complicated repeating shapes like the hexagonal pattern of *grid cells* (Moser et al., 2008). For a comprehensive review of NRES modalities identified in neuroscience, see (Bicanski and Burgess, 2020).

Neural state is very different from the monolithic state of RL. Analogous to separate cells representing coordinates of one Euclidean space, separate NRES modalities reflect

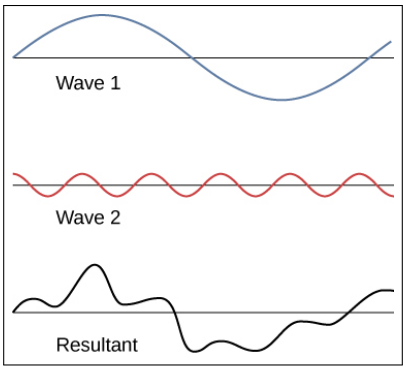

Figure 1: Two simple sinusoidal functions can be combined to a complex function by superposition. (Ling et al., 2016)

different aspects of the navigational state. The receptive fields of NRES neurons have a systematic increase from the dorsal to the ventral pole of the hippocampus (Fyhn et al., 2008; Kjelstrup et al., 2008; Solstad, 2009), allowing for NRES maps of multiple resolutions in parallel. The fully distributed representation of state thus allows for learning state representation by individual receptive fields, for different NRES resolutions and across NRES modalities in parallel. The monolithic Markov state of RL (Sutton and Barto, 2018), on the other hand, could explain difficulties for robot interaction learning (Kaelbling, 2020). The most protruding difference between AI and neural state representation lies in the distributed nature of NRES. We now explore how this can be emulated for RL systems.

Decomposing the Prediction Problem

The purpose of an *agent* in reinforcement learning is to establish a proper behavior as defined by a reward signal. The agent improves behavior based on two intertwined aspects of experience: (1) The *prediction problem* for learning the value of visiting states or state-action pairs as defined by the environment representation, and (2) The *control problem* for selecting the most appropriate action based on the value as learned by the prediction problem. In this section, we expand on the concept of the prediction problem by considering the value function as a potential field across orthogonal reward signals.

Let Orthogonal Value Functions (OVFs) be value functions of the state space $\mathbb{S}$ that adhere to mutually exclusive reward signals in $\mathbb{S}$. A relevant analogy would be to think of the value function as a potential field between different sources of energy. With multiple forces working on an object, the resultant work can be found as a linear combination of components. Similarly, a set of independent reward functions in $\mathbb{S}$ acting on agent value function can form a basis for agent value function in $\mathbb{S}$. NRES with mutually exclusive receptive fields is a good candidate for independent reward signals; with the place cell as our leading example, it is simple to visualize how agent position activates receptive fields and OVFs. Each *learner* has a simple reward shape, with a positive reward of $\mathbb{R} = +1$ upon activation of the corresponding NRES cell and $\mathbb{R} = 0$ otherwise. A separate learner form the OVF according to reward signals as defined by mutually exclusive receptive fields of $\mathbb{S}$.

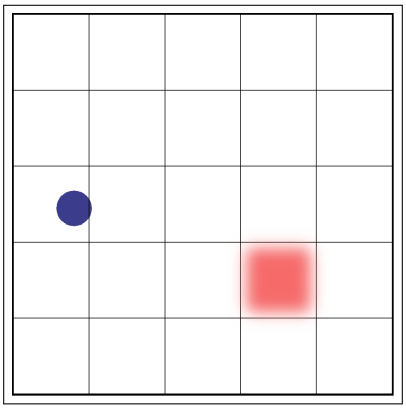

Figure 2: An agent in $N5$ allocentric place-cell representation of Euclidean space: An $N5$ representation involves that each axis is divided into 5 equal intervals. A learner could, for example, form the OVF toward cell $(4, 4)$, with a reward signal defined by the activation of the corresponding NRES cell. The reward function of this particular learner is illustrated in red for feature $s_R \in \mathbb{S}$. The current parameter configuration of the agent defines from which $s \in \mathbb{S}$ this NRES modality's value function is extracted.

Let there be K individual learners, one for every receptive field of an NRES representation $\mathbb{S}$. With mutually exclusive receptive fields, the set of learners in $\mathbb{S}$ can be considered an orthogonal basis of the value function in this representation. Value functions of $\mathbb{S}$ can be expressed as a linear combination of OVFs formed by the K learners, allowing a neoRL agent to synthesize a range of behaviors. The challenge of learning apt behavior now reduces to learning priorities between policies expressed via OVFs. Estimating scalar values based on supervised samples is a well-studied field in machine learning. However, for the sake of clarity, static priorities defined by the associated reward is used.

The Control Problem by Superposition

The motivation for learning the value function is ultimately to form an effective policy for the challenge at hand. A simple challenge in Euclidean space can be for the agent to move to one particular position, activating feature s_x. If learners use Q-learning to establishing a potential that contributes to the *Q-field* of the agent, the next action can be chosen by

$$a = \text{argmax}_a Q_{tot}(s, a)$$

where Q_{tot} is the resultant Q-field of the current situation. With a single learner as input to the agent value potential,

the agent's prediction problem becomes equivalent to that of the single learner, and the mechanism surrounding the value function of the agent simplifies to that of a monolithic agent.

For slightly more interesting challenges, multiple rewards can be expressed in the decomposed NRES representation. Each learner can be said to represent one consideration in this environment, learning the value function related to activating the corresponding NRES cell. When multiple considerations have priority, the superposition principle allows the Q-field to form over relevant OVFs.

$$Q_{tot}(s,a) = \sum_{i \in \mathbb{S}_R} Q_{\mathcal{L}i}(s,a) \tag{2}$$

where $\mathbb{S}_R$ is the set of NRES cells associated with reward and $Q_{\mathcal{L}i}(s,a)$ represent learner $\mathcal{L}_i$'s value component. The K learners in the full features set can thus be considered to be *peer learners* for the task of navigating the environment representation.

$$\mathbb{S}_R = \{s \in \mathbb{S} \;\middle|\; |\mathbb{R}_s| > 0\}$$

An elegant approach would be to consider rewards to be linked to elements of interest in the environment rather than allocentric features: Let an *Element of Interest* (ξ_i) be an instance in the environment associated with a reward. Assume for now that the priority and Euclidean parameter configuration of every element of interest in the set $\mathbb{E} = \{\xi_i\}$ is provided by the environment. Any parameter configuration is possible to map uniquely to the mutually exclusive NRES feature map $\mathbb{S}$. With element i's importance w_i proportional to the reward associated with the element activating feature s, the corresponding peer learner's contribution to the Q-field becomes:

$$Q_{tot}(s,a) = \sum_{i \in \mathbb{S}_R} w_i Q_{\mathcal{L}i}(s,a) \tag{3}$$

Isolating rewards that comes from elements of interest, i.e. abstaining from utilizing timestep rewards or other shaped rewards, the set of rewarded states is defined by the set of NRES cells occupied by an element of interest ξ_i.

$$\mathbb{S}_R = \{s \in \mathbb{S} \mid \exists \xi_i \in \mathbb{E},\ \xi_i \in s\} \tag{4}$$

Note that an element of interest can be any element associated with a reward in a particular state set representation, decoupling the prediction problem in an environment from the rewards of one task. Experience expressed by distributed Q-fields is more general than monolithic value functions; In the neoRL approach, moving rewards or changing agent priorities during an agent's life-time does not require retraining the agent.

Experiments

Algorithms in RL learn behavior by interaction with the environment, making the environment defining for the outcome of any RL experiment. Numerous environments exist to highlight challenges for state-of-the-art reinforcement learning agents. Learning autonomous navigation in allocentric space does not seem to get much attention, as finding appropriate test-environments can be difficult. Preferably, an environment for autonomous real-world navigation learning is represented by continuous allocentric coordinates and with a complexity that requires reactive navigation. Real-time execution would be a plus since it limits the amount of training data available to the agent to a realistic order of magnitude. Physical systems generally depend on temporal aspects like inertia. Most of these qualities can be found in Karpathy's WaterWorld challenge.

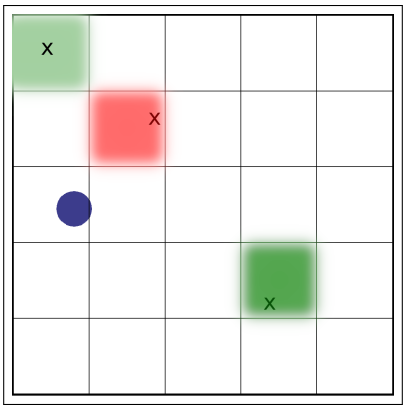

Figure 3: Element of Interest (EoI) activates desires for allocentric features according to their importance: An EoI situated in feature $(4,4)$ makes this desirable with 1.0 , another positive EoI activates feature $(1,1)$ with priority 0.5 , as represented by a green with lower saturation. An aversive element located in feature $(2,2)$ activates the corresponding learner with a negative weight $w_i < 0$.

WaterWorld

Karpathy's WaterWorld challenge as implemented in Pygame learning environment(PLE) (Tasfi, 2016) is an environment with a continuous 2D resolution, inertia dynamics and external considerations referred to as *creeps*. Creeps move with a constant speed vector, reflected when hitting a wall. Creeps have a demeanor, as illustrated by color: green creeps are desirable with [+1] reward, and red creeps are repulsive with [-1] reward upon capture. When the agent captures a creep, a new one is initialized with a random speed, position, and demeanor – causing a chaotic scenario that requires reactive navigation. When all green creeps have been captured, the board is restarted with an accompanying [+5] reward. In all experiments, a constant number of 8 creeps have been used, as illustrated in Figure 4. We find the allocentric PyGame implementation (Tasfi, 2016) of WaterWorld appropriate for RL research for real-time navigation autonomy. However, the environment is listed as unsolved

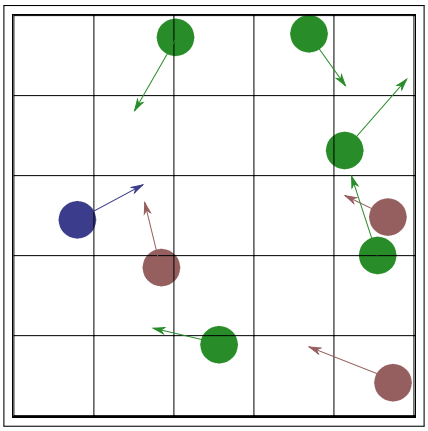

Figure 4: NRES $N5$ representation of Element-of-Interest (EoI) in the *WaterWorld* environment. Each EoI and the location of the agent represented in the PlaceCell NRES modality. Red and Green represent the demeanor of each creep, whereas Blue represents the current agent location. In addition, arrows have been drawn to illustrate the current speed vector of each element.

(OpenAI, 2020) – making comparisons to alternative solutions difficult.

Instantaneous information regarding elements-of-interest (EoI), i.e., the position and demeanor of each creep, is provided by the environment. Demeanor defines the reward associated with the creep, crucial for priority w_i associated with EoI i by equation 3. Positions are represented in 2D allocentric coordinates from the environment, allowing for extracting $\xi_i \in \mathbb{S}$ for the Place Cell NRES modality of EoI i. Basal actions affect the agent by accelerating it in the cardinal directions, [N, S, E, W].

Allocentric Position Modality, Single layer: Our primary assumption is that the agent value function in effect can be considered a potential field across OVFs, pulling the agent toward the next decision. Our first experiment explores to what degree the superposition principle holds for the value function of individual learners. We compare the accumulated score of neoRL agents based on single-res NRES to Brownian motion, i.e., an ϵ-greedy policy with $\epsilon = 1.0$. Under the convention used in Figure 4, where $N5$ signifies an NRES map with $5x5$ tiles, five different resolutions are explored from $N10$ to $N90$. All experiments were conducted over 150.000 time-steps for each neoRL agent.

Allocentric Position Modality, Multiple resolutions: Our second experiment explores how integrating experience across multiple state spaces affect neoRL performance. An interpretation of the progressive increase for receptive fields in the ventral direction of the hippocampus is that different NRES maps exist with different resolutions. We adopt this view in experiment 2, where we let the neoRL agent combine value function across multiple NRES state representations. In this experiment we assess whether the neoRL agent is capable of forming apt policies by integrating experience across multiple state spaces. We compare the proficiency of a multi-res neoRL agent that learns over $\{N3, N7, N23\}$ NRES state spaces to three single-res agents by $N3$, $N7$, and $N23$ NRES. The neoRL agent layout is illustrated in Figure 5. Prime numbers are used as the resolution for each layer, minimizing the potential for overlapping boundaries. The resulting 587 learners in the multi-res agent learn in parallel by off-policy learning. In this setup, the contribution of each learner is inversely proportional to the size of its receptive field.

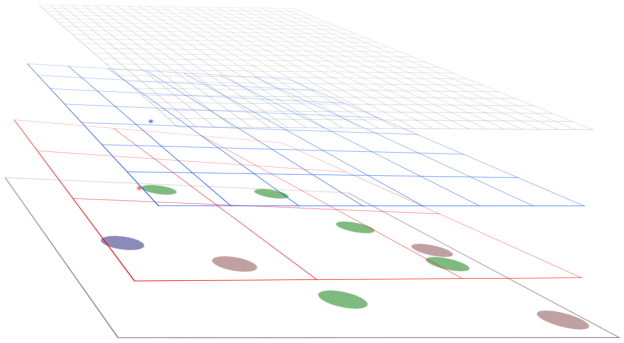

Figure 5: Illustration of multiple state representation in the decision agent, where each tile represent the objective of its respective learner. [Red] N3 representation [Blue] N7 representation [Black] N23 representation.

One approach of measuring the proficiency of the agent is as the per-timestep average reward across parallel runs. We are interested in real-time learning efficiency and initialize a neoRL agent with no priors at the beginning of each run. A per-timestep average across 100 independent runs provides information about the transient timecourse in navigation capabilities. Note that every run starts with a separate neoRL agent with no prior experience. All experiments are conducted on an average desktop computer, with one run taking somewhat under one hour on a single CPU core.

Results

Results are reported as real-time execution of agents as they learn, without any previous experience at the task. Reported resolution for each experiment adheres to the convention from Figure 2, dividing each axis of the Euclidean space into N steps. The x-axis of all plots represents the number of time steps since the beginning of a run, i.e., the real-time execution in time-steps since initiation of the agent.

Allocentric Position Modality, Single layer A distributed representation of the Markov state is plausible for neoRL agents. Figure 6 shows the accumulated score of

neoRL agents with NRES Place Cell representations from $N10$ to $N90$. All neoRL agents perform better than control. Brownian motion seems incapable of achieving a single board reset since the accumulated score fluctuates around 0 for the length of the experiment. All neoRL agents are capable of accumulating a significant amount of experience, verifying that OVF can function as a basis for synthezing successful behavior.

A strong correlation between NRES resolution and proficiency at the task can also be observed in Figure 6. The immediate proficiency at the task can be seen from the steepness of the curve. Agents based on lower NRES resolution initially learn quicker than agents with higher NRES resolution. However, neoRL agents based on lower NRES resolutions seem to saturate at a lower proficiency. For these particular runs, with 8 creeps and during a 150.000 time step interval, the $N50$ representation appears to achieve the highest score. Although this number is task-specific, it is worth noting how all neoRL agents are comparable in learning speed. Despite $N70$ NRES having almost 50 times the dimensionality of $N10$[1], the two neoRL agents based on these representations are comparable in learning. This effect requires further attention.

Allocentric Position Modality, Multiple resolutions Combining the value potential from multiple representations of state can significantly increase navigation performance. The transient proficiency of the neoRL agent in the four experiments, $N3$, $N7$, $N23$, and multi-res $\{N3, N7, N23\}$, is presented in Figure 7. Each curve is the result of a per-timestep average over 100 independent runs. These results verify without any doubt that neoRL agents benefit from combining experience across multiple NRES feature sets. With the algebraic sum of the per-timestep proficiency of the three mono-res agents shown in grey, we see that the multi-res neoRL agent learns quicker, to higher proficiency, than the sum of its parts.

The superposition principle for behavior across state spaces seems to alleviate the curse of dimensionality: The almost 6-fold increase in the number of states (from $7^2 = 49$ to $3^2 + 7^2 + 23^2 = 290$ states) resulted in a 3.5-factor increase in received reward without increasing training time. Figure 7 shows that learning happens as fast or possibly a little faster for the multi-res agent than for the $N7$ mono-res agent. This effect could be defining for real-world interaction learning and requires further attention.

Discussion

Navigation autonomy is plausible in real-time by RL agents with an emulated neural representation of space. NRES-Oriented RL (neoRL) agents are possible due to developed theory on orthogonality in the value domain, allowing for behavior synthesis across multiple learners.

Whereas neural systems are capable of autonomous navigation, modern technology is not. The most protruding difference between these systems is how state is represented. Digital RL systems require a monolithic state concept, whereas neural systems work by patterns of activation. The Markov state in RL holds enough information to uniquely define the probability distribution of the next state (Sutton and Barto, 2018). The Markov decision process works well with deep function approximation, and RL agents supported by deep learning have mastered a selection of board games. However, deep RL agents require much training, do not generalize, and are neither incremental nor compositional (Kaelbling, 2020). With deep RL appearing to struggle with real-world interaction learning, we have looked elsewhere for inspiration. Evidence suggests that Neural Representation of Euclidean Space (NRES) represent Euclidean coordinates by activation patterns on the per-neuron level. An NRES set $\mathbb{S}$ with mutually exclusive receptive fields provides a set of orthogonal reward signals of $\mathbb{S}$. Utilizing these signals as reward signal for independent learners, the set of Orthogonal Value Functions (OVFs) form a basis for any reward function of $\mathbb{S}$. Experiments verify that NRES-Oriented RL (neoRL) agents are capable of forming skilled navigation while learning.

Considering this work as a plausibility study for neoRL navigation, we see at least three important directions for further study. Firstly, a thorough mathematical analysis on the relevance of orthogonality could be key for proper understanding of neoRL capabilities. Specifically, deriving the equations for how singular reward functions cause orthogonal value functions can cause a better understanding of behavior synthesis. In experiment 2, we have seen how different state-space representations of the same parameter set can improve performance. We believe the same to be possible for state spaces across different parameter spaces. Secondly, the priority w_i in Equation 3 remains static in this work but allows for a dynamic weighing of OVF based on importance. Directly learning the association between element i and global reward $\mathbb{R}$ would make neoRL learning comply to the reward hypothesis, and be an important continuation of this work. Lastly, all experiments conducted on the neoRL framework have yet been with the WaterWorld environment. The WaterWorld represents a quite general task in a highly general Euclidean space across undefined parameters. Many would find it more interesting with a tangible demonstration in a more specific Euclidean space, e.g., navigation of the joints' angles in a robot manipulator task. A most important next step would be to demonstrate neoRL navigation for other Euclidean spaces, e.g., for maritime autonomy, (learned) autonomous driving, or for adaptive control of robot manipulators.

[1]The $N10$ representation is comprised of 100 receptive fields, whereas the finer $N70$ resolutions have 4900 receptive fields.

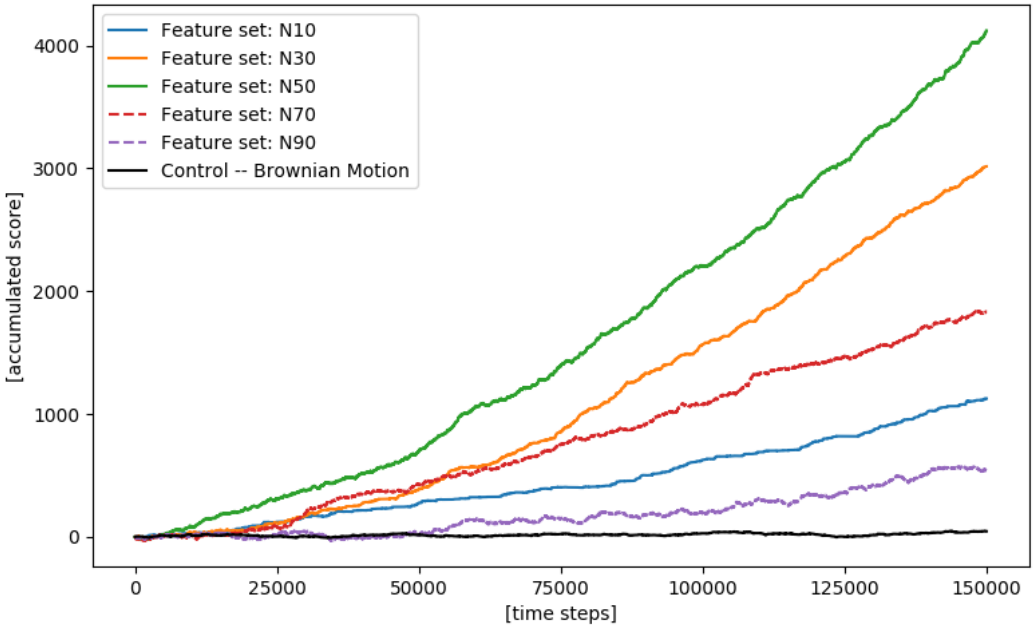

Figure 6: Accumulated Reward by peer agents with elements of interest for runs with grid coding resolutions, $N10 - N90$ over 150.000 time steps. Brownian motion in black is believed to be comparable to a first run of an untrained Deep RL agent.

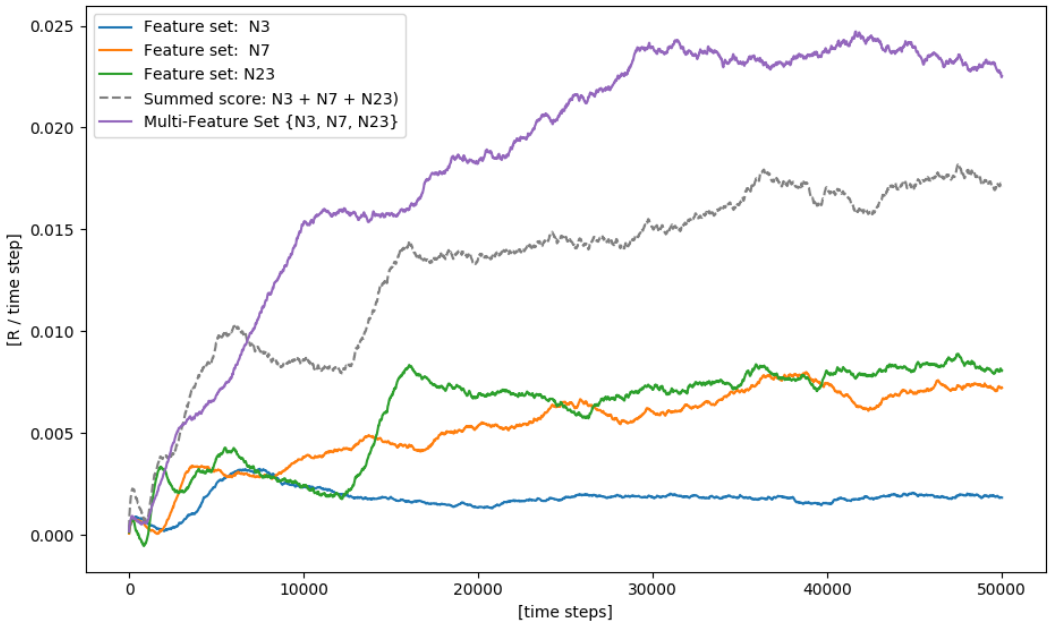

Figure 7: The neoRL agent is capable of incorporating experience from multiple state sets for navigation. A neoRL agent with experience from all three layers seen in Fig. 5 (purple) performs better than neoRL agents based on the individual NRES layer (blue, orange, green). The grey line represents the algebraic sum of the mono-res agents, highlighting that the multi-res neoRL agent performs better than the sum of its parts. Each curve is a presentation of the per-timestep average of 100 independent runs.

References

Abbeel, P., Coates, A., Quigley, M., and Ng, A. Y. (2007). An application of reinforcement learning to aerobatic helicopter flight. *Advances in neural information processing systems*, 19:1.

Bicanski, A. and Burgess, N. (2020). Neuronal vector coding in spatial cognition. *Nature Reviews Neuroscience*, pages 1–18.

Fyhn, M., Hafting, T., Witter, M. P., Moser, E. I., and Moser, M.-B. (2008). Grid cells in mice. *Hippocampus*, 18(12):1230–1238.

Fyhn, M., Molden, S., Witter, M. P., Moser, E. I., and Moser, M.-B. (2004). Spatial representation in the entorhinal cortex. *Science*, 305(5688):1258–1264.

Heinrich, J. and Silver, D. (2016). Deep reinforcement learning from self-play in imperfect-information games. *arXiv preprint arXiv:1603.01121*.

Høydal, Ø. A. (2020). *Allocentric vector coding in the medial entorhinal cortex*. Unpublished PhD thesis, Kavli Insitute of Systems Neuroscience / Center of Neural Computation.

Kaelbling, L. P. (2020). The foundation of efficient robot learning. *Science*, 369(6506):915–916.

Kandel, E. R. and Tauc, L. (1965). Heterosynaptic facilitation in neurones of the abdominal ganglion of aplysia depilans. *The Journal of Physiology*, 181(1):1.

Kjelstrup, K. B., Solstad, T., Brun, V. H., Hafting, T., Leutgeb, S., Witter, M. P., Moser, E. I., and Moser, M.-B. (2008). Finite scale of spatial representation in the hippocampus. *Science*, 321(5885):140–143.

Kober, J., Bagnell, J. A., and Peters, J. (2013). Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274.

Latombe, J.-C. (2012). *Robot motion planning*, volume 124. Springer Science & Business Media.

Ling, S. J., Sanny, J., Moebs, W., Friedman, G., Druger, S. D., Kolakowska, A., Anderson, D., Bowman, D., Demaree, D., Ginsberg, E., et al. (2016). *University Physics Volume 1*. OpenStax.

Moser, E. I., Kropff, E., and Moser, M.-B. (2008). Place cells, grid cells, and the brain's spatial representation system. *Annu. Rev. Neurosci.*, 31:69–89.

O'Keefe, J. and Dostrovsky, J. (1971). The hippocampus as a spatial map: Preliminary evidence from unit activity in the freely-moving rat. *Brain research*.

OpenAI (2020). OpenAI, Snake v0.

Ramón y Cajal, S. (1911). *Histologie du système nerveux de l'homme et des vertébrés*, volume 2. A. Maloine.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. (2017). Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*.

Solstad, T. (2009). *Neural representations of Euclidean space*. PhD thesis, Kavli Insitute of Systems Neuroscience / Center of Neural Computation.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

Sutton, R. S., Modayil, J., Delp, M., Degris, T., Pilarski, P. M., White, A., and Precup, D. (2011). Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 761–768. International Foundation for Autonomous Agents and Multiagent Systems.

Tasfi, N. (2016). Pygame learning environment. https://github.com/ntasfi/PyGame-Learning-Environment.

Taube, J. S., Muller, R. U., and Ranck, J. B. (1990). Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis. *Journal of Neuroscience*, 10(2):420–435.

Tesauro, G. (1994). Td-gammon, a self-teaching backgammon program, achieves master-level play. *Neural computation*, 6(2):215–219.

Van Seijen, H., Fatemi, M., Romoff, J., Laroche, R., Barnes, T., and Tsang, J. (2017). Hybrid reward architecture for reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 5392–5402.

Whitlock, J. R., Sutherland, R. J., Witter, M. P., Moser, M.-B., and Moser, E. I. (2008). Navigating from hippocampus to parietal cortex. *Proceedings of the National Academy of Sciences*, 105(39):14755–14762.

Wiener, N. (1948). *Cybernetics: Control and communication in the animal and the machine*. Wiley.

Wiering, M. A. and Van Hasselt, H. (2008). Ensemble algorithms in reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(4):930–936.