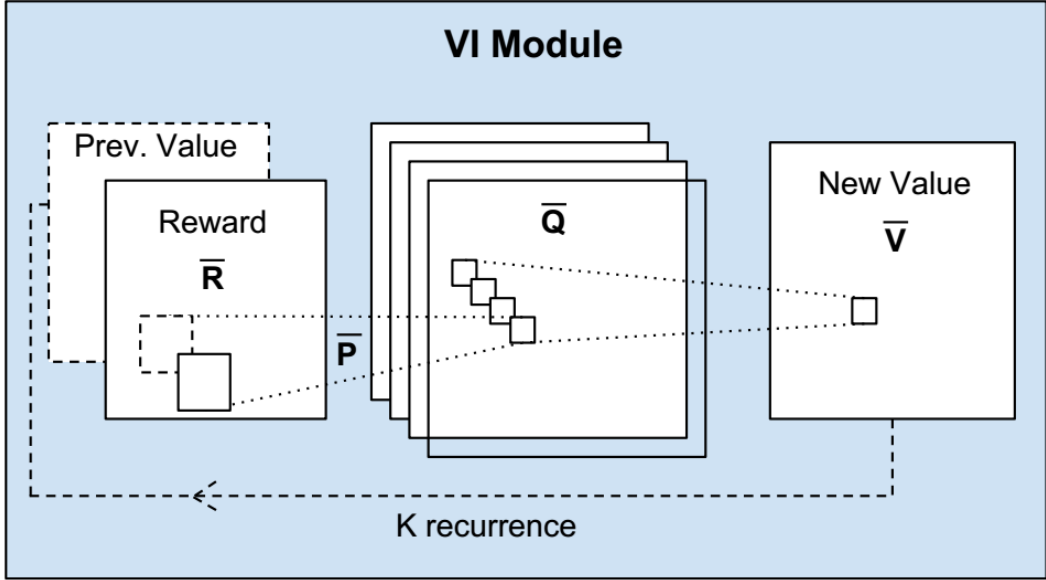


Title	NIPS2016 Value Iteration Networks: Aviv Tamar, UC Berkeley		
Abstract	<div>> They introduce the value iteration network (VIN): a fully differentiable neural network with a ‘planning module’ embedded within.</div> <div>> The VIN can be used to predict outcomes that involve planning-based reasoning, such as policies for reinforcement learning.</div> <div>> Key to the VIN is a novel differentiable approximation of the value-iteration algorithm, represented by a CNN, and trained using standard backpropagation.</div> <div>> Evaluate the VIN based policies on discrete and continuous path-planning domains, and on a natural-language based search task.</div>		
Introduction	Methods	Pros	Cons
	Classic deepRL	<div>> Neural network is trained to represent a policy: mapping <observation> to <action>, the goal is get a policy has good long-term behavior.</div>	<div>> Supervised learning is one-step decisions making. However, RL needs some form of planning.</div> <div>> Recent deepRL model are inherently <i>reactive</i>, and lack explicit planning computation. The success of <i>reactive</i> policy in sequential problems is duo to the <i>learning algorithm</i>, the algorithm is trained to select actions that have good-long term consequences in its training domain.</div>
	Model RL		Require system identification, infeasible for complex real system.
The cons example of Recent deepRL (reactive strategy)	<div><div></div><div></div></div> <div>Figure 1: Two instances of a grid-world domain. Task is to move to the goal between the obstacles.</div> <div>People's expectations of such type of games are after training a policy to solve several instances of this problem with different obstacle configuration, the policy would generalize to solve a different, unseen domain configuration. However, in experiment, with standard CNN-based network, we can easily solve a set of such</div>		

		maps, however, the policy trained do not generalize to new tasks outside this set, because they do not understand the goal-directed nature of behavior.
	Proposed Method	<p>> They proposed a NN-based policy that can effectively learn to plan. Their model, termed a value-iteration network (VIN), has a differentiable 'planning program' embedded within in the NN structure.</p> <p>> VIN could be trained model free, without requiring explicit system identification. Moreover, the errors in VINs can be mitigated by training the network end-to-end.</p>
Backg round	Value Iteration	The goal in an MDP is to find a policy that obtains high rewards in the long term $V_{n+1}(s) = \max_a Q_n(s, a) \quad \forall s$, where $Q_n(s, a) = R(s, a) + \gamma \sum_{s'} P(s' s, a) V_n(s')$.
	Convoluti onal Neural Networks (CNN)	
	Reinforce ment Learning and Imitation Learning	<p>> When MDP transitions or rewards are not know in advance, planning algorithms cannot be applied. In this cases, a policy can be learned from either <i>expert supervision -IL</i> or <i>by trail and error -RL</i>.</p> <p>> In imitation learning, learning a policy becomes an instance of supervised learning</p> <p>> In reinforcement learning, the agent can act and observe the rewars and the state transitions its action effect.</p>
The Value Iteratio n	Overview of the Model	<p>Notation</p> <p>M : denotes the MDP of the domain for which we design our policy.</p> <p>\bar{M} : unknown, contains the useful information about the optimal policy in the original task M .</p>

Network Model	<p> $\bar{s} \in \bar{\mathcal{S}}, \bar{a} \in \bar{\mathcal{A}}, \bar{R}(\bar{s}, \bar{a})$ Are the states, actions, and rewards and transitions in \bar{M} </p> <p> $\bar{R} = f_R(\phi(s)), \bar{P} = f_P(\phi(s))$ are depends on the observation in M. </p> <p> θ is the parameters of f_R, f_P, π_{re} </p> <p> > This section mainly focus on how to use the planning result \bar{V}^* with in the NN policy π. The approach is based on two important observation. The first is that the vector of $\bar{V}^*(s)$ encodes all the information about the optimal plan in M. The second is that the MDP has a local connectivity structure, it only depends on a subset of the values of $\bar{V}^*(s)$, which refers to an attention module. </p> <div data-bbox="423 720 1459 1436" data-label="Diagram"> <p>The diagram illustrates the Value Iteration Network (VIN) architecture. It consists of the following components and data flow:</p> <ul style="list-style-type: none"> Observation: A dashed box containing $\phi(s)$. VI Module: A large box containing: <ul style="list-style-type: none"> Inputs: f_R and f_P feed into a dashed box containing \bar{R} and \bar{P}. Plan on MDP \bar{M}: A solid box that takes \bar{R} and \bar{P} as input and outputs \bar{V}^* (in a dashed box). Attention: A solid box that receives input from the Observation box and the VI Module (specifically from the \bar{V}^* output). Reactive Policy: A solid box that receives input from the Attention box and the Observation box. It outputs $\pi_{re}(a \phi(s), \psi(s))$. Intermediate Representation: A dashed box containing $\psi(s)$, which is the output of the Attention module. </div>
The VI Module	<p> > Each channel in the convolution layer corresponds to the Q-function for a specific action. </p>

		<p>> Each convolution kernel weights correspond to the discounted transition probabilities.</p> <p>> Representing VI in this form makes learning the MDP parameters and reward function natural- by back propagating through the network.</p> 
	Value Iteration Networks	<p>> The VIN is based on the general planning-based policy defined above, with the VI module as the planning algorithm.</p> <p>> In order to implement a VIN, one has to specify the state and action spaces for the planning module \bar{S}, \bar{A}</p>
Experiments	Overview	<p>The goal in these experiments:</p> <p>> Can VIN effectively learn a planning computation using standard RL and IL algorithms?</p> <p>> Does the planning computation learned by VINs make them better than reactive policies at generalizing to new domains?</p>
	Grid-World Domain	
	Mars Rover Navigation	
	Continuous Control	
Conclusions		