

Project Plan: Reinforcement Learning in Latent Space

We, the authors

October 11, 2018

1 Context and Motivation

In **Reinforcement Learning** (RL), machine learning problems are modelled as a sequence of actions taken by an agent in some environment to maximize a total reward. Instead of learning from a dataset, the agent builds knowledge about the environment by exploring the effect of its behaviour. Since such interaction can be cost-intensive in real-world (or physical) applications, it is desirable to pre-train agents on a simulated task and afterwards generalize the obtained knowledge to the real task.

For this reason, recent research on reinforcement learning has increased attention on **transfer learning** (TL). In TL, an agent learns to do a *source task* and uses its knowledge in a before unseen *target task* to perform to a reasonable level with minimal additional training. In its more extreme forms, TL is known as one- or zero-shot-learning, where only a single training step is allowed in the target task - or none at all (**goodfellow2016deep**). In RL, one of the key challenges of applying TL is the inter-task alignment of states and actions. While some work tackles this issue with hand-crafted solutions (**taylor2007cross**), it is desirable to find solutions for an automatic transfer. (**taylor2008autonomous**) do this for instance by automatically mapping states and actions between tasks based on a short exploration phase. Another approach, which will be used in this project, is to learn in a common state (and possibly action) space, into which all tasks can be translated.

Deep Learning (DL) has been successfully applied to a variety of problems in machine learning research and got increasing attention over the last two decades (**goodfellow2016deep**). For instance, **hessel2017rainbow** combined several improvements to Deep Q-Learning in one algorithm and achieved state-of-the-art performance with an agent playing multiple Atari games. **finn2018model** use deep learning to create a state-of-the-art few-shot transfer learning algorithm that can be used for different problems, including reinforcement learning. Although DL is applicable to classical tasks such as regression and classification - and may henceforth be used as a policy learner in RL - it is particularly useful for learning representations in a latent space. For example, convolutional neural networks (CNN) can be used to break down visual input into features that model higher-level information. Sequences of variable length, such as natural language, can be embedded using recurrent neural networks (**goldberg2017neural**). Another architecture for representation learning is the so called *autoencoder* (**hinton2006reducing**). These neural networks learn to first reduce the dimensionality of their inputs. They then reconstruct the original sample from the low dimensional representation. The low-dimensional representation in the middle layer often proves to be useful as a representation for different applications as it encodes the relevant information of the input space into a latent space.

Motivation Using DL, it may be possible to find state and action representations for multiple tasks that allow joint learning and generalization to unseen tasks. This is useful for several reasons. As stated previously, it facilitates learning in costly tasks. Take, for example, the training of a robot

playing soccer. If the agent needs to learn its behaviour entirely from physical play, the training process could only be run in real time. A simulation modelling the environment closely could be run much faster and, if done correctly, the learning is generalizable to the original task. Tasks that seem infeasible to train become solvable. Going further, knowledge gathered from playing soccer should be useful in other ball sports or even physical activities in general. If we aim to develop agents capable of performing any task in a given domain¹, we would expect it to use experience from one task in other tasks if they are at least partially applicable. For that, the different state spaces (and their representations) need to be abstracted into some latent space, as can be done with DL. In this latent space, all tasks may benefit from the experience in other tasks and only need to adjust to the specific requirements of the new situation. For instance, soccer and handball share the objective of bringing a ball into a goal as well as basic movement patterns and cooperation strategies. However, they differ in the way the ball is handled.

Objective In this project, we aim to design a learning framework in which multiple RL tasks can be trained in the same latent space. The resulting knowledge should be generalizable to unseen tasks, whose training gets kick-started or at least sped up. An agent capable of playing multiple and partly unseen Atari games will be developed. It is hypothesized that Variational Autoencoders (VAE) can serve as a technique to convert the different states of different games into a latent space. The following research questions are posed:

Research Question I Can a joint latent state space be learned for multiple different tasks from visual input using Deep Learning?

Research Question II Can the state space of unseen tasks be translated to the latent space without additional training on the new task?

Research Question III Can a single agent learn multiple tasks when using the latent space to represent states during Q-Learning?

Research Question IV Does the policy network trained on the source tasks significantly improve the training of unseen tasks in the latent space?

Research Question V Does the policy learned in the source tasks allow an agent to perform to a descent level on the target tasks without additional training?

Research Question VI Can we identify crucial information encoded in the latent space? If yes, what are those pieces of information that are encoded in the latent space and what behaviour do they entail?

2 Social Impact

While reinforcement learning proved its effectiveness in diverse areas ranging from chemistry ([zhou2017optimizing](#)) to games ([silver2017mastering](#)), solving complex problems typically requires costly computational resources. This could be a factor blocking individuals or smaller organizations from utilizing such techniques. However, if knowledge learnt on one task can be reused on other related ones, this entrance barrier could be greatly lowered. Since this research project focuses on transferring knowledge

¹Such as sports or any class of activities that rely on a common ground of skills (e.g. motor skills).

across tasks, the proposed method, if proven successful, could make reinforcement learning techniques more accessible and usable.

A concern about the topic of this project, however, could be that knowledge transferring is one of the first steps towards “general” artificial intelligence. Several prominent scholars and industry leaders warned about potential reinforcement-learned agents that can manipulate or control their reward signals (**russell2016research**). However, since existing research on transfer learning in reinforcement learning were generally validated on simple simulated physical games or Atari games, we do not consider this risk sufficiently realistic to be relevant.

3 Concepts and approach

The development of the project will require the use of different deep learning algorithms in order to tackle the final goal. Tasks like the feature extraction from the Atari frames, the generation of the latent space and the finding and training of the agent policy will be handled by deep learning techniques.

When dealing with Atari games, the problem of constructing features-vectors from raw pixels can be addressed using **Convolutional neural networks (CNNs)**. CNNs, are a class of neural networks designed to process “grid-like topology data” (**goodfellow2016deep**) like images or sounds signals. CNNs allow to “abstract” features from the raw pixels matrix using an operation called *convolution*, a special kind of linear operation. Each feature will be represented by a *filter*: a vector of weights and the respective bias that are incrementally adjusted to “learn” how to correctly extract that specific feature (e.g. a square shape). Furthermore, by using gameplay frames as input for the CNN, we assure that each task to later embed in the latent space has the same data structure for the state space.

In order to generate the latent space, **autoencoders** will be used. An autoencoder is a neural network formed by two parts: the encoder, that converts the input to a dense and smaller representation and the decoder, that rebuild the input from the representation. The main application for autoencoders is the dimensionality reduction (**Hinton504**), where the encoder learns to preserve the meaningful attributes of the input and generates a lower dimensional representation that is saved in the **latent space**.

As we work with different tasks, the latent space will be the place where we aim to save a generalized representation of the state spaces. To create a continuous latent space, that could allow transfer among task, a **variational autoencoder (VAE)** will be used. This kind of neural network use probability distributions to describe each feature, allowing interpolation and random sampling in the data. Moreover, VAEs are trained with **gradient based methods** which gives a better control over the latent space representation (**goodfellow2016deep**) and can be used to generate representations with disentangled factors (**2016arXiv160605579H**).

To achieve transfer learning, the agent will learn from the generalisation of the states stored in the latent space using deep reinforcement learning techniques. This algorithms will be used with the objective of creating an agent that will be capable of detecting similar states and situations that occurs on tasks and act in a similar way in each of them.

4 Deliverables

What we want to deliver is an agent that we trained using reinforcement learning. The training environment that we will use is the OpenAI Gym (**openai gym**). This enables us to implement different approaches with varying difficulty. A relative easy approach will be to train the agent on sinusoidal tasks such as a cartpole task, a mountain car task or a pendulum task. For this, we will use the

internal state representation at first, later, if possible, transforming these problems to a visual state representation.

A second deliverable will be an agent that is able to play Atari games using pixels as input. Although the OpenAI Gym framework also supports this, it will be harder to implement because we will need a lot more time to train an agent. This renders debugging very time consuming. Furthermore, none of us has experience with pixel based input data, which will make constructing the needed neural network difficult.

Since we want to use a latent space for transfer learning we will need to deliver a representation learner. This learner will be based on an autoencoder model, more precisely on a variational autoencoder. The states that were used for training the VAE should be encoded in the learned representation, which in turn could be reused for future work.

Most importantly, we will deliver a method for transfer learning and an implementation of it. We will show that it is possible to learn from previous tasks and use knowledge from those to achieve a better performance in another related task. We will show this using sinusoidal tasks as well as Atari games. These trained models will be available for demonstration purposes.

Furthermore, the results will be compared using different metrics, showing their different kind of strengths such as jumpstart, convergence and overall performance. Additionally, a qualitative analysis of the learned representation will be given. This analysis will investigate what kind of features are still present in the latent space and how an agent can learn from these representations.

5 Time Management

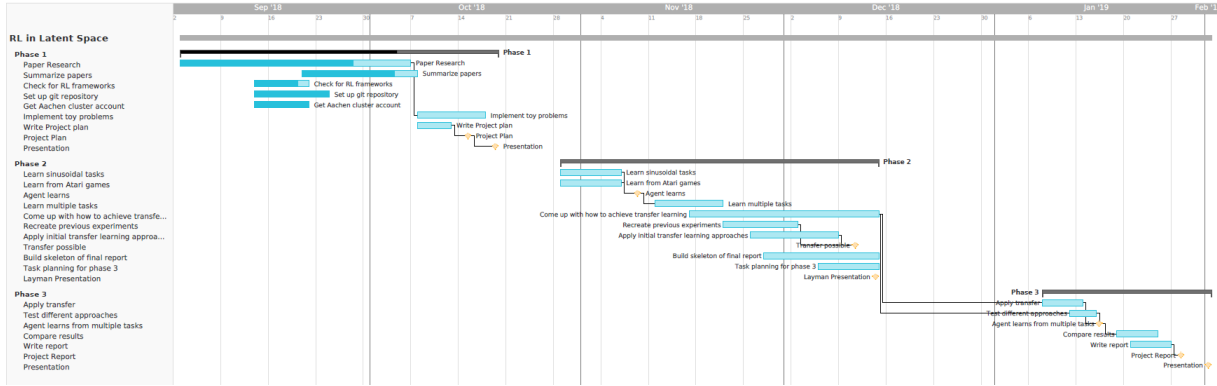
In order to tackle the project efficiently we dedicate a lot of time of the first phase into researching previous papers that already looked into transferring learning between different tasks. Furthermore, we will look for useful reinforcement learning frameworks which can be used to test our method. To verify that sufficient knowledge of these frameworks is present we will have to implement some toy problems which should be done by the end of phase 1.

In phase 2 we will split our group up into two groups to focus on different tasks that the agent will be trained on. One of these approaches is an easier implementation of learning as a fall-back if the other one does not work out. The main focus in this phase lies on coming up with a method to transfer learning between different tasks. Towards the end of this phase we plan to recreate existing transfer learning experiments and maybe even implementing our own first attempts. Furthermore, a skeleton for our final report that we have to hand in in phase 3 is drafted.

Phase 3 will then focus on applying and implementing our ideas to transfer learning. After achieving this, we will compare our results with already existing ones and start writing our final report.

5.1 Gantt Chart

To illustrate, we created a Gantt chart to show our task planning.



6 Risk Analysis

We distinguish between technical and organizational risks, where the former involve those related to the approach chosen to tackle the research problem, whereas the latter refer to those arising from the project team itself or related parties. The two types of risks are listed in table 6.1 and 6.2 respectively. In addition to the descriptions and contingency plans, justifications for the costs and likelihoods of the risks are provided in table 6.3.

6.1 Technical Risks

ID	Risk Description	Contingency Plan	Cost	Likelihood
1	The team cannot reproduce the results reported in selected papers on time.	Provide explanations, and substantiate with experiment results.	+	+++
2	The team cannot implement a working RL-based Atari-playing agent on time.	Focus on transfers across sinusoidal tasks (inverted pendulum, mountain car, cart pole) and explore a larger variety transfer methods.	+++	++
3	The proposed method does not bring improved performance (no effect or “negative transfer”) on the interested tasks pairs / groups.	Investigate the reasons of the failure to transfer. Compare with scenarios where transfer is successful.	++	+++

6.2 Organizational Risks

ID	Risk Description	Contingency Plan	Cost	Likelihood
4	One team member has to be absent from the team for long due to uncontrollable factors like severe illness or family circumstances.	Reduce the scope of the project. When in the phase of implementation, reduce the number of alternative models. When in the phase of experiments, reduce the number of experiments and prioritize those with the highest chance of yielding insightful results.	+++	+
5	Downtime in the Aachen computing cluster makes it impossible to run experiments as scheduled.	Rent GPU-enabled virtual machines on Microsoft Azure or Amazon Web Services (AWS) with the free student credits.	++	+

6.3 Cost and Likelihood Explanation

ID	Cost Explanation	Likelihood Explanation
1	+: Reproduced experiments results, regardless whether the same as in the original publications, can still be used as a comparison with our experiment results.	+++ : It is likely that not all hyper-parameters or initialization conditions are explained in the publication, making the experiment results difficult to reproduce.
2	+++ : With a large number of experiment scenarios (Atari games) ruled out, the experiments will be limited to sinusoidal tasks.	++ : Although no team member had experience with game control based on raw pixels, there are existing tutorials and Git repositories that accomplished this task.
3	++ : Negative results without well-interpreted results would render this research unsuccessful.	+++ : It may be challenging to apply suitable techniques to achieve transfer across our experiment tasks, as transfer learning in reinforcement learning is still an area with open issues.
4	+++ : The absence would cause a 20% drop in manpower.	+: Factors causing the long-term absence of a teammate are rare.
5	++ : Experiments results may be delayed, impacting project progress.	+: Long-lasting downtime of the Aachen cluster is unlikely.