Transfer of Status Report

Adam Kosiorek[1]

***Abstract*— Abstract goes here.**

I. Introduction

Computers are often praised for being better than humans at things that are inherently hard for humans, yet they are also criticised for being considerably worse than humans at things that are very easy for humans . While some algorithms perform well without being trained on huge amounts of data , they are often domain specific . Often to increase performance, much more complicated model and much larger amounts of data are required. It has been recently shown that performance in computer vision tasks is logarithmic in the size of the dataset.

Add reference(s).

Add reference(s).

- where is it?

Humans, on the other hand, are known to be extremely good at learning. We can often learn and generalise well after seeing only a single labelled data point. Some argue , that this can be explained by learned feature representation. Our brains are hard-wired to build specific representations of the world and are exposed to millions of images every day, constantly inferring truths about the world. That is in contrast to modern machine learning algorithms, which are typically trained only once and they are deployed with their parameters frozen. Moreover, modern algorithms are typically trained from a cold start, without any prior knowledge of the world.

Add reference(s).

Add reference(s).

Finally, human cognition often involves using time-dependencies in the data . This contradicts current computer vision practice, where the majority of algorithms is used for inference from single data points. While reasonable in some problem domains, sequential character of data can be leveraged in other domains for higher performance, more consistent model outputs, better learning with better generalization. I would argue that even in domains where inference from single data points is required, training of those algorithms could benefit from sequential nature of data.

Add reference(s).

Add reference(s).

II. Related Work

Things to write about:

- Sequence prediction: text, motion, videos
- Predictive Coding
- Semi-supervised learning: learning by association and ladder networks
- behaviour learning by RL: maze navigation, locomotion patters
- unsupervised learning: AIR

A. Sequence Prediction

Sequence prediction has been a long standing problem in machine learning. Early works on neural networks used next-step prediction of chaotic time-series as a benchmark . More recently, next time-step prediction has been addressed in the context of natural language processing, where the goal was to predict the next word or a character conditioned on some previous texts. It is often argued that this time of prediction requires vast contextual knowledge and often requires understanding of some abstract rules about the workings of the world. It is also considered for motion prediction, specifically in the joint-angle space, where the goal is to predict joint configuration of a given skeleton conditioned on some post motion of this skeleton . A slightly different take on the subject is presented by , where the task is to predict motion of an arbitrary number of objects even when they are occluded.

Add reference(s

Add reference(s

Add reference(s

Next-timestep prediction is done with a plethora of different methods, many of which share the central component in the form of a recurrent neural network. Depending on the domain, the RNN can implement different state transition, evolve over different timescales, use several layers, use attention or employ domain-specific structure of the hidden state e. g., feature maps for visual inputs.

Add reference(s

Another characteristic consistent across domains is that this task can be learned in an unsupervised, or father self-supervised, fashion. No explicit ground-truth is needed to learn to predict a sequence, since the ground-truth is the sequence itself, albeit shifted.

Add reference(s

- d
t
i

Yet another trait particular to next time-step prediction is that the model maintains some hidden state which describes the state of the world. We can argue that whenever the prediction of the model is perfect (the prediction matches the next time-step perfectly), the hidden state perfectly reflects the state of the world and therefore does not have to be updated. None of the aforementioned approaches make use of that observation, and therefore in every of the presented cases the model has to learn to compute discrepancies between the new observation and the hidden state of the world and to update the hidden state by corrected those discrepancies only. This is possible, as proved by the satisfactory performance of the presented approaches, but that means that all next-timestep prediction settings have a common structure that has been neglected so far. Below I describe the idea of predictive coding, and how frameworks following this approach address the limitation of usual sequence predictors.

[1] Oxford Robotics Institute, Dept of Engineering Science, University of Oxford.

B. Predictive Coding

Add reference(s).

The idea of predictive coding dates back at least to the Kalman filter . Kalman filter is an instance of a Gaussian linear system used for estimating the state of the world. It first computes a prediction of the world state at the next timestep and then, when it gets hold of the observation at that timestep, it updates its prediction. It also estimates noise covariance matrices for the prediction and update steps and their respective contributions are proportional to the inverse covariance matrices.

Add reference(s).

Friston argues that predictive coding can be implemented as energy minimisation and that energy minimisation can well explain learning in the brain and many neuroscientific phenomona that would otherwise remain quite puzzling to neuroscientists. He suggests that the computational architecture of the brain forms a hierarchical system, where each layer constantly predicts the output of the lower levels of hierarchy in a fully Bayesian fashion. It gives rise to surprise, which is the negative log-likelihood of the inputs under the predictive distribution of the model. Friston argues that surprise leads to error normalisation w. r. t. prediction uncertainty, which can be interpreted as attention.

It is possible to frame the next-timestep prediction in terms of an auto-regressive neural network. Indeed, it has been addressed in a number of papers (**Lotter2016**; **Canziani2017**). These models use the mean-squared error between the prediction and the corresponding input to update their hidden state. It leads to an ill-posed problem since there are many futures possible. The prediction, which is a maximum-likelihood estimate of what might happen, is only a single instantiation thereof.

C. Semi-supervised and Unsupervised Learning

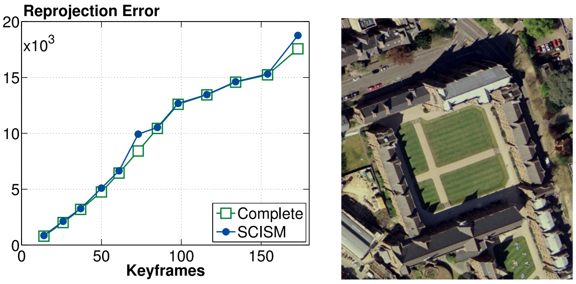
Next time-step prediction and predictive coding both fall in the category of unsupervised learning, where the aim is to learn the probability distribution over sequences of observations $\mathbf{x}$, namely $p(\mathbf{x}_{1:T})$. Some of the approaches do so by assuming latent variables $\mathbf{z}$ the explain the data and learn the joint distribution of $p(\mathbf{x}, \mathbf{z})$ instead. Any type of auto-encoder falls within that category.

1) autoencoders, denoising, constrastive
2) ladder networks
3) deep belief nets
4) vae
5) learning by association

III. Research Proposal

Things to write about:i

- Predictive Coding: next-frame prediction with VAEs, it's normalisation behaviour, inner feedback loop for corrections
- Model-based RL: sample efficiency, using a policy to improve learning speed of the perception module, learn a policy in the absence of a goal
- Unsupervised object tracking & detection



(a) (b)

Fig. 1: Two figures wrapped in a table: (a) a graph and (b) a college.

IV. Conclusions

Conclusions go here.

Appendix

General theme: Representation learning for sequential data.

I'm interested in:

- timeseries
- unsupervised learning
- predictive coding
- learning abstract concept and ideas

What I'd like to do is:

I'd like to leverage unsupervised learning for time-series to learn abstract concepts that describe the world, and more importantly, its evolution. I would like to be able to:

- predict how the world evolves
- find out whether doing so in a probabilistic way has any benefits over deterministic approaches, e.g. is multi-modality of a probabilistic solution helpful
- see how imposing structure on the predictions affects representation learning, e.g. air-style unsupervised object tracking; using a policy to either minimise or maximise surprise to improve learning speed, predictive accuracy or both

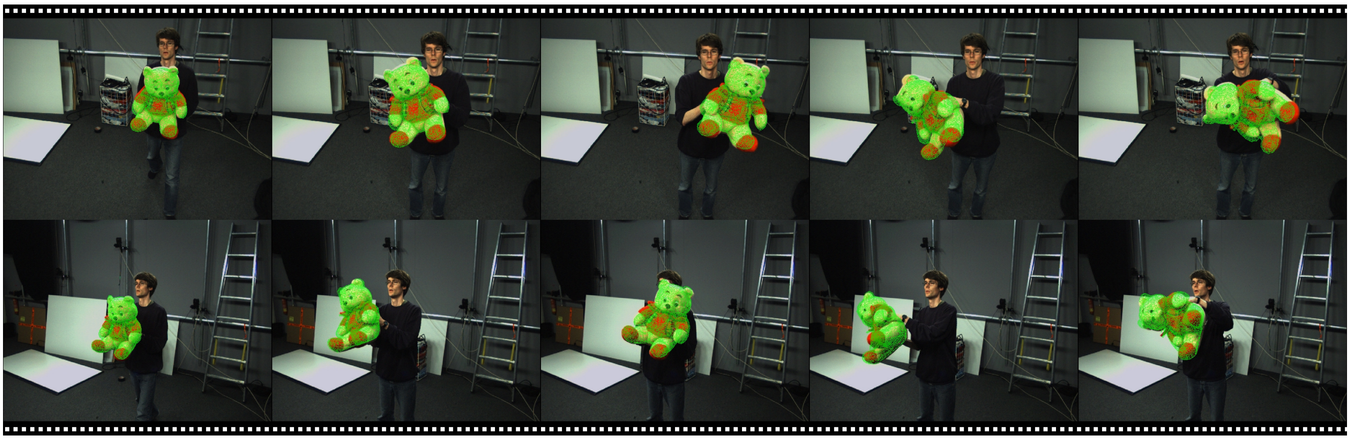


Fig. 2: A figure that spans both columns is produced using the figure⋆ environment.