PolyChord

CHAPTER

ONE

INTRODUCTION TO POLYCHORD AND ITS NEURAL NETWORK OPTIMISER TOOL, POLYNET

PolyNet is a neural network optimiser based on the cutting-edge maths tool *PolyChord*, invented by a leading Cambridge University Astrophysics team, Prof Mike Hobson, Prof Anthony Lasenby and Dr Will Handley in 2017.

1.1 Neural Networks

A neural network is a mathematical model mapping observations onto predictions. The work-flow for neural network design and usage typically takes the form:

1. Collect a set of labelled training data.
2. Design/choose a network architecture.
3. Train the network by determining the 'best' set of neuron weights.
4. Evaluate the network's quality, and if necessary return to a refined version of step 2.

PolyNet provides unique advantages for both Steps 3 and 4.

Step 3. is usually performed using a gradient-based optimiser such as TensorFlow. Optimising a neural network may be treated as a high-dimensional model fitting problem, which PolyNet approaches in a *Bayesian fashion*, using PolyChord as it's engine. Step 4 is typically executed by using cross-validation techniques. PolyNet allows one to use Bayesian evidences to assess the quality of the network.

1.2 PolyChord

The underlying engine PolyChord is a "hands off" Bayesian optimizer and evidence calculator, representing the cutting-edge of nested sampling technology. PolyChord is widely used in astrophysics and cosmology, providing more accurate and reliable answers in comparison with all existing tools.

PolyChord is uniquely specialised for navigating high-dimensional multi-modal posteriors with complicated shapes and degeneracies, which typically cause industry-standard optimisers to founder. Moreover, PolyChord's evidence calculation explicitly quantifies how good a model is relative to other architectures.

For academic papers detailing the original algorithm (PolyChord Lite) please see:

- PolyChord: nested sampling for cosmology – arXiv:1502.01856, MNRAS 450(1) L61-L65,
- PolyChord: next-generation nested sampling – arXiv:1506.00171, MNRAS 453(4) 4384-4398

Applying PolyChord Lite to neural network training:

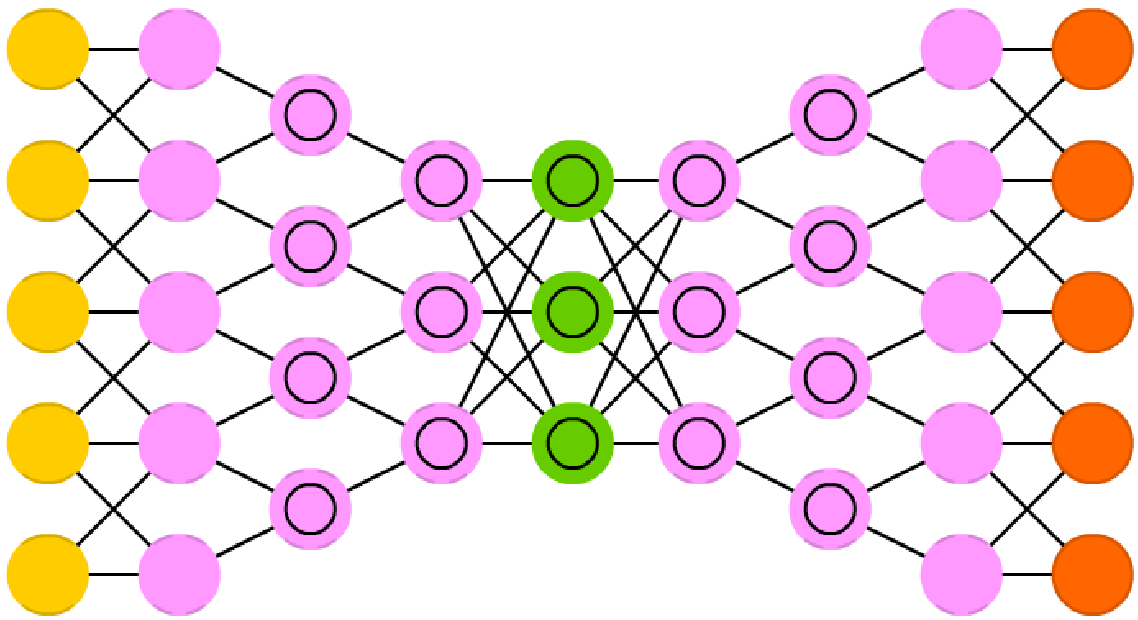

Fig. 1: An example network architecture. Source: Asimov institute.

- Bayesian sparse reconstruction: a brute-force approach to astronomical imaging and machine learning – arXiv:1809.04598

PolyChord has been used in a host of analyses across astronomy and particle physics Google Scholar

1.3 The PolyNet approach

Fundamentally, PolyNet is a Bayesian neural network sampler and evidence calculator. The advantages it presents over existing technology are two-fold: improved sampling techniques and evidence calculation.

1.3.1 Nested sampling for training Bayesian neural networks

Nested sampling has a unique ability to navigate a-priori unknown curving degeneracies and multimodality in posterior distributions. These are precisely the kind of challenges found in Neural network training.

PolyChord represents the state of the art in high-dimensional nested sampling, putting it head-and-shoulders above existing approaches in its ability to train neural networks. We expect PolyNet to outperform existing approaches both in speed and quality of training though speed is not our main focus due to our specialisation in the field of 'offline training': models are trained offsite once-and-for-all and then port trained models used for their implementation.

1.3.2 Evidence calculation for assessing network architecture

PolyChord is capable of computing Bayesian evidences in high dimensions. This is essential for evaluating the Bayesian quality of a given neural network architecture in fitting the data. PolyChord computes the Bayesian evidence as a matter of course when fitting models (technically model fits are a by-product of its evidence computation).

One can therefore use PolyChord's outputs to navigate network architectures. Bayesian evidences may be used to construct a full likelihood loop that fits models, and then favourably adjust the network architecture. The final product

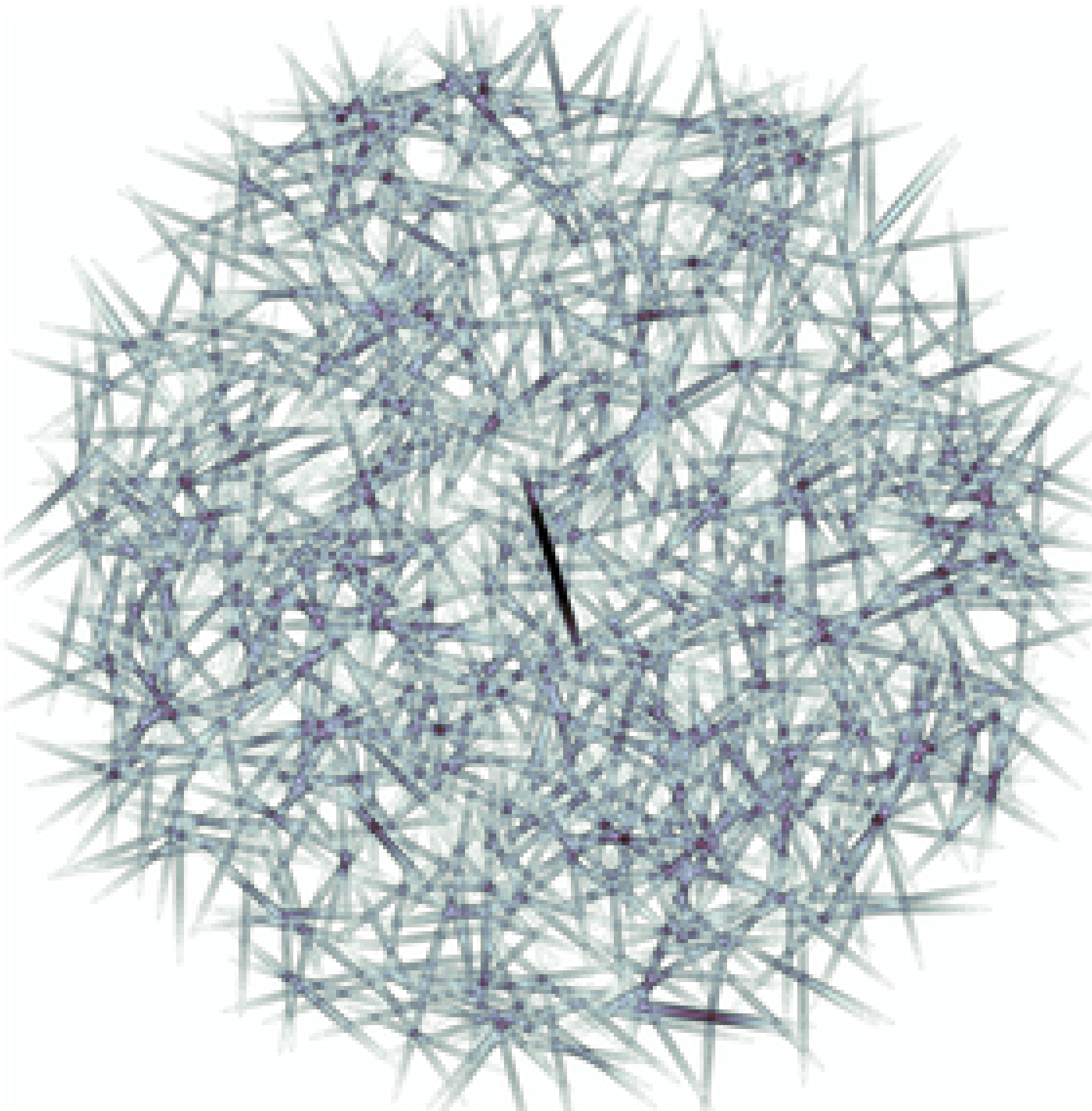

Fig. 2: Searching for a needle in a haystack: PolyChord is uniquely capable of navigating complicated optimisation surfaces

is a two-step Markov Chain Monte Carlo (MCMC) algorithm. PolyChord is used to MCMC fit a given network, and then a step in 'the network space' is made, prioritising steps toward better networks. The final product is a weighted sum of Bayesian fits to the network, as well as greater insight into what network architectures are preferred.

1.4 Unique Advantages in Addressing Neural Net Training

The unique point of difference in PolyChord is that we have simultaneously both a better fitting algorithm and a genuine evidence calculation. Current attempts at evidence calculation in this field are based on brutal approximations, which has been the primary barrier for anything becoming a more standard tool. Our main competitor here is Google's neural net optimiser TensorFlow that quickly evaluates gradients and perform variations on gradient descent such as stochastic gradient descent, momentum update, and adaptive learning rate. Such methods are optimised for speed of training rather than quality, and are beset by issues such as poor regularisation (an inability to determine the quality of fit) and false-minima (optimising to a locally good models, but missing the global best one). By looking at the whole space, PolyNet delivers greater efficiency and accuracy.

CHAPTER

TWO

MAXIMISERS VS. SAMPLERS

Finding a good fit for a neural network requires combatting two issues: Multimodality, and overfitting. Multimodality occurs when there is more than one "good" fit, and is particularly important where there are multiple "best" fits.

Model fitting is performed by a variety of approaches which fall into two categories: *Maximisers* and *Samplers*.

2.1 Maximisers

Maximisers fit models by finding the single 'best fit' parameters of the model. For example:

- Gradient methods
- Stochastic optimisation
- Genetic algorithms

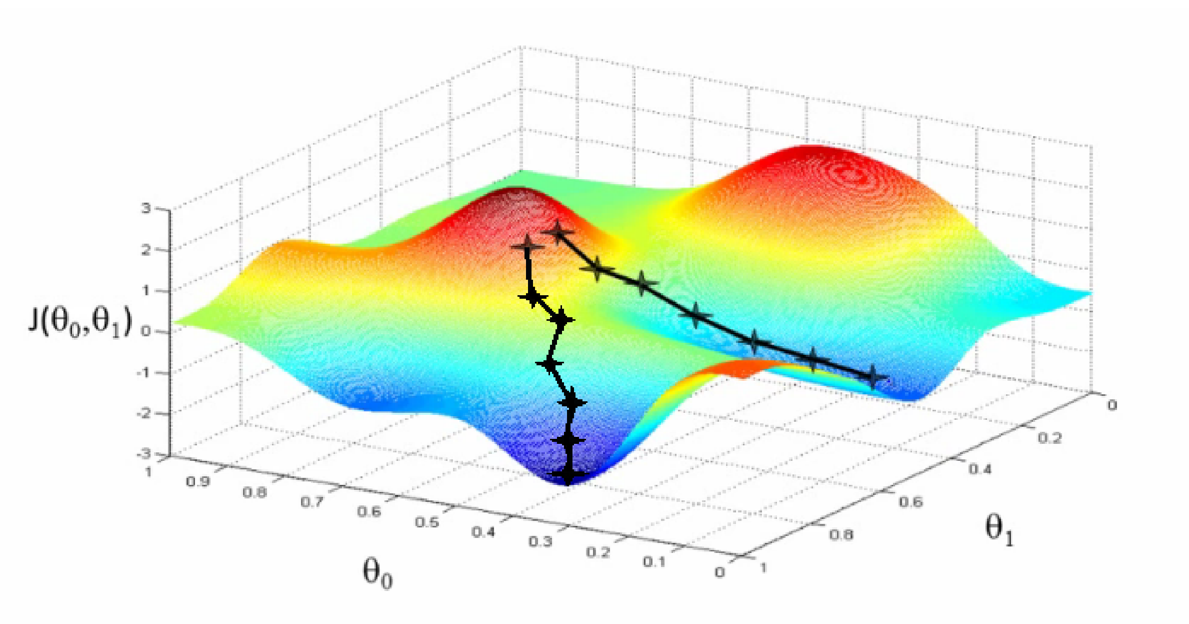

Fig. 1: Example of gradient ascent for non-convex loss function (such as a neural network), with two parameters θ_0 and θ_1. Source: Andrew Ng.

In general, such methods are prone to getting stuck in local peaks and may miss the global maximum. However, in some situations there may be multiple peaks of equal quality. In such cases, choosing a single peak can correspond to confining your fit to a single mode of operation. More advanced methods use multiple peaks simultaneously to find the optimal fit.

2.2 Samplers

Samplers fit models by finding a collection of 'most typical' parameters of the model, and typically do so using a Markov Chain Monte Carlo (MCMC) approach. For example:

- Metropolis Hastings
- Nested Sampling
- Simulated Annealing

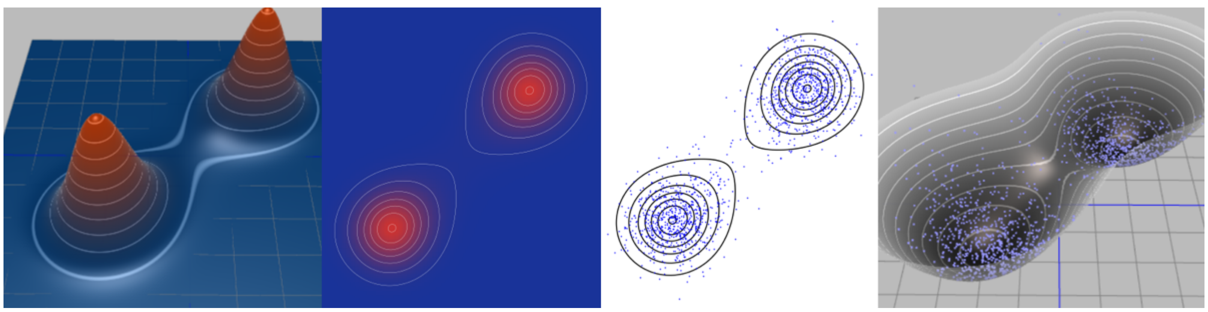

Fig. 2: Example of samples drawn from a bimodal posterior distribution. Source: Alex Rogozhinikov.

Sampling is advantageous for two reasons. First, a perfect sampler will explore multimodal distributions correctly. Second, sampling a function naturally combats over-fitting and allows quantification of errors in your analysis and the fidelity of your fit.

Maximisers attempt to combat overfitting via methods like regularisation and dropout. However, the reasons and intuition behind why such methods work are obscure, dubious, and a little ad-hoc. Moreover, optimising maximiser-based neural networks requires trial-and-error attempts in selecting and adjusting hyperparameters, which is not the most effective and qualitative approach to machine learning.

For the reasons above, the use of PolyChord represents a preference of sampling over maximising.