Population-Based Training (PBT) for Neural machine translation (NMT)

Introduction

Neural Networks showed great success in most of the domains they are used in starting from audio/images classification, playing video games, as well as, machine translation, and other tasks Natural Language Processing (NLP) related. Building and training a neural network for a specified task is easy, however, optimization of networks is still a hard task to do, specially, when dealing with General Adversarial Networks (GANs) or Double SARSA/Actor Critic based Deep Reinforcement Learning techniques. As there are a lot of hyper parameters to tune, this tuning can be done using "Random Search" and/or "Manual Tuning".

Hyperparameters are the set of parameters that define how the model will be structured. It can be thought of it as searching a parameters space to find the best parameters. i.e. our aim is to find the optimal set of hyperparameters as per task. Generally, this process can be broken down into the following:

1. Defining and building the model
    - Task specific
2. Define a range of possible values
    - Recurrent Neural Network LSTM based
        - Dropout
        - Recurrent Dropout
        - Batch Size
        - Number of Epochs
    - Decision Trees classifier
        - Number of trees
        - Maximum depth
3. Define an evaluation criteria

Optimization Techniques

Optimization techniques can be categorized into 2 classes:

1. Parallel Search: many parallel optimization processes, trains multiple networks with different set of hyperparameters, for example Grid Search & Random Search.
2. Sequential Optimization: it follows same paradigm as parallel search for few iterations of optimization and get the output of these iterations utilizing them to improve NNs performance gradually, for example Manual tuning & Bayesian Optimization.

- Sequential optimization will be better compared to parallel, however, it is not feasible for long optimization processes.

Brief about different optimization approaches

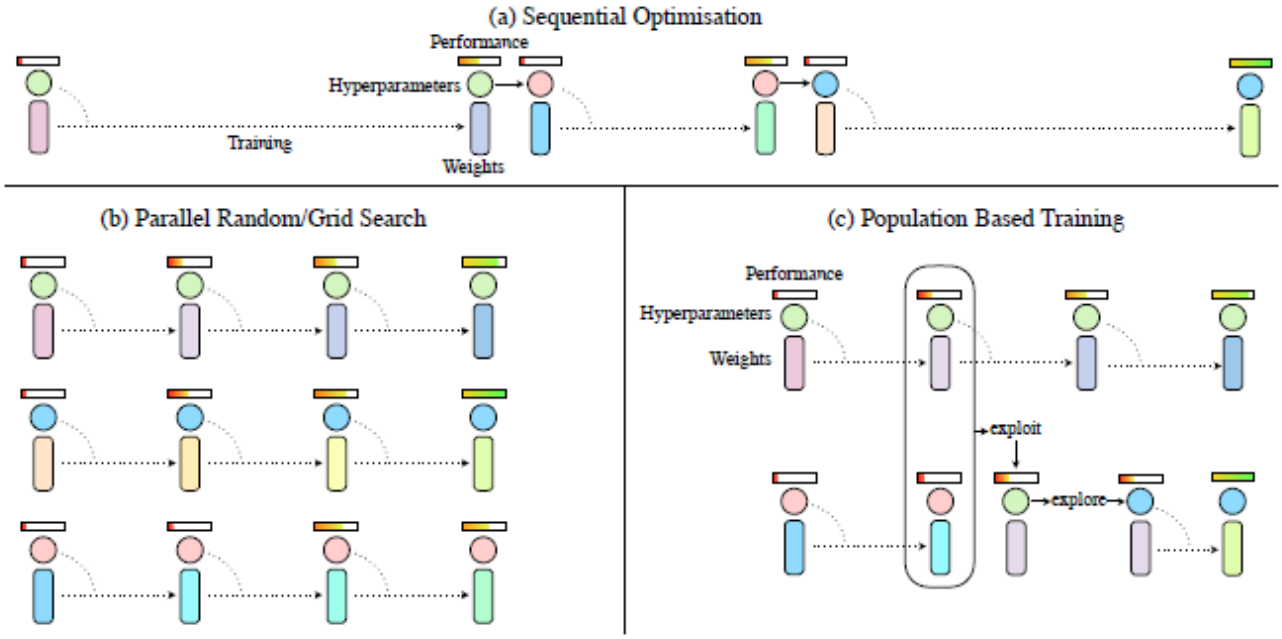
1. Grid Search: Most basic tuning method, which is based on building and training models as per each set of hyper parameters, which is very time and resources consuming, e.g. if we have a model and 100

combinations of hyperparameters, this means we have to wait for training of 100 models and then pick the best. This illustrates the suffering of Grid Search if the number of parameters grow.
2. Random Search: several neural networks are built and trained asynchronously, and the best one in terms of performance is selected. But, all NNs are trained in the same time regardless of how promising the network will be at the end. It is easy to notice, that some NNs are not as good and will consume computational resources that can be saved.
3. Manual Tuning: Deep Learning practitioner has to guess a set of hyper parameters, and train the NN. This is a basic iterative optimization method which takes a lot of time, which can be invested in something else, which increase the chances of finding best fit params comparatively higher; as the random search ends up optimizing parameters without any aliasing.
4. Bayesian Optimization: Unlike what was mentioned earlier, it keeps track of past evaluation results, which are used to form a probabilistic model mapping hyperparameters to a probability of a score on the objective function

---

Population Based Training of Neural Networks

It basically is an optimization technique that aims to get the best of both worlds parallel and sequential optimization. So, it trains multiple networks at the same time, and also able to use fewer computational resources in comparison with random/grid search. It leverages information sharing across a population of concurrently running optimization process and allows for online transfer of hyperparameters between members based on performance.



Definition

It starts like parallel search by randomly sampling hyperparameters and weights initialization of the model. Nevertheless, each training runs asynchronously evaluate its performance periodically. If the model is under-performing, it will exploit the rest of the population by replacing itself with a better performing model, and it will explore new hyperparameters by modifying the better model's hyperparameters, before training is continued. The result is a hyperparameter tuning method that while very simple, results in faster learning, lower computational resources, and often better solutions.

So, Simply, it is an asynchronous optimization algorithm which effectively uses a fixed computational budget to jointly optimize a set of neural networks (will be referred to as "population" later) and their hyperparameters to maximize the performance.

PBT discovers a schedule of hyperparameters settings rather than following the generally sub-optimal strategy of trying to find a single fixed set to use through the whole course of training

---

References

1. Population Based Training of NN paper- arxiv
2. Deepmind Blog
3. Optimization techniques Blog I
4. Optimization techniques Blog II