Population Based Training of Neural Networks

Max Jaderberg, Valentin Dalibard, Simon Osindero
DeepMind

arXiv:1711.09846

Information & Intelligence System Lab.
Sang Heon Lee
lawlee1@naver.com

2019/01/11

Introduction

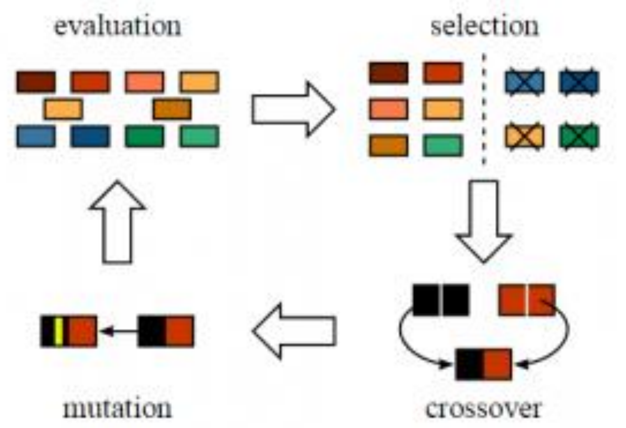
- Hyperparameter of Model
  - Hyperparameters must be properly tuned to fully unlock network performance
  - Tuning process is computationally expensive ➔ Hyperparameter search methods

- Hyperparameter search methods
  - Parallel search
    - Training models with diff. hyperparameters in parallel, select the best one
    - Require a lot of computational resources

  - Sequential optimization
    - Using information obtained from earlier training runs
    - Require a lot of time

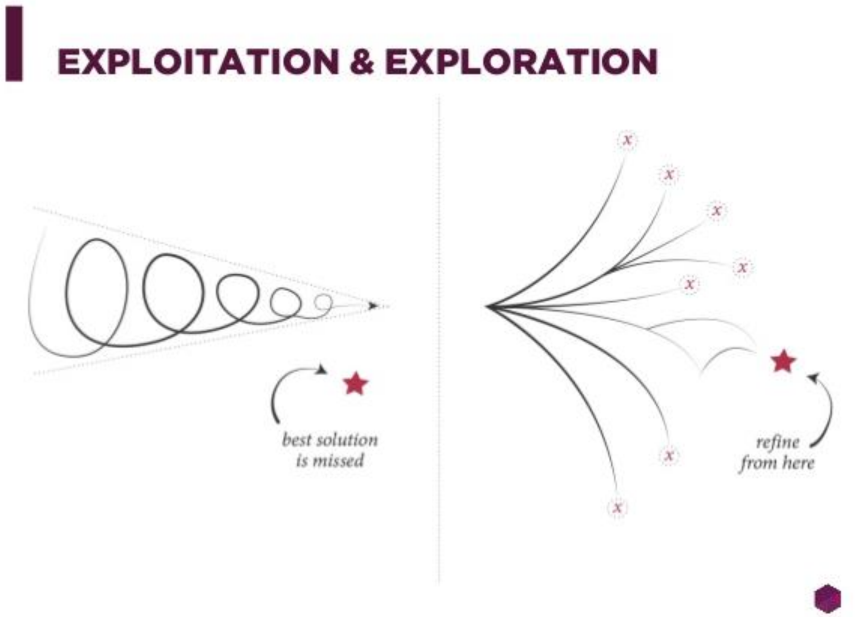
Introduction (Cont'd)

- Present simple hyperparameter searching methods
  - Bridges between parallel search and sequential optimization methods
  - *Population Based Training (PBT)*
  - Based on genetic algorithm
    - Information sharing across a population
    - Transfer **parameters** and **hyperparameters** between members of the population



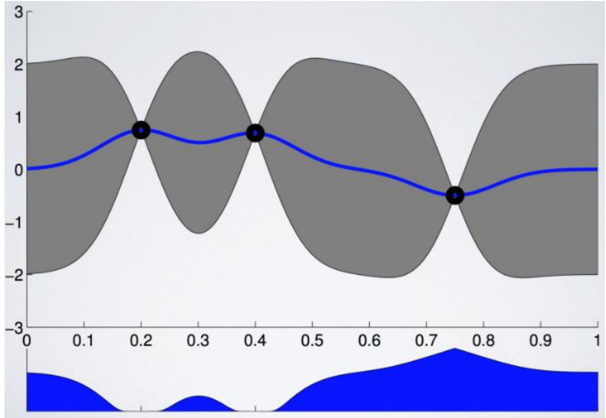
Related Works

- Optimization : Exploitation vs. Exploration
  - Exploitation : choose the best point until now
  - Exploration : choose a new point to get more information
  - Trade-off relations : the proper control of these two behaviors is core



Related Works (Cont'd)

- Sequential optimization : Bayesian Optimization
  - Utilize the information of previous trial points
  - Goal is finding $\mathbf{x}^\star = \arg\max_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$
  - Approximate expensive function f using a probabilistic surrogate model



Algorithm 1 Bayesian optimization

for $n = 1, 2, \ldots$ **do**
    select new $\mathbf{x}_{n+1}$ by optimizing acquisition function α

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}} \alpha(\mathbf{x}; \mathcal{D}_n)$$

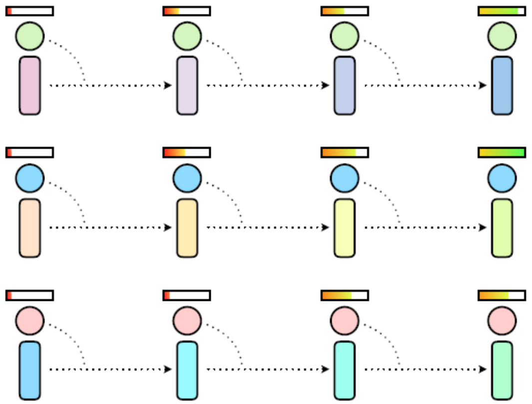
    query objective function to obtain y_{n+1}
    augment data $\mathcal{D}_{n+1} = \{\mathcal{D}_n, (\mathbf{x}_{n+1}, y_{n+1})\}$
    update statistical model
end for

- Require **a lot of time** to find the best solution

Related Works (Cont'd)

- Parallel search : Random search

(b) Parallel Random/Grid Search

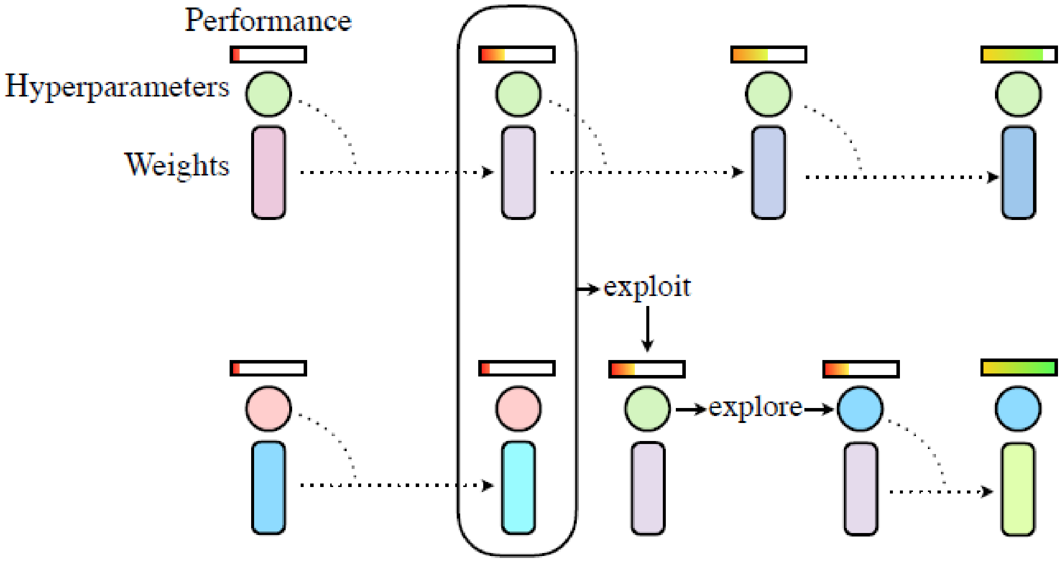


  - Waste computation on bad hyperparameters
  - Fails to utilize the **information of history**

  - *Hyperband* (Li et al., 2016)
    - Allocate more budget to more promising hyperparameter configurations
    - Has same problem as random search

Proposed Method

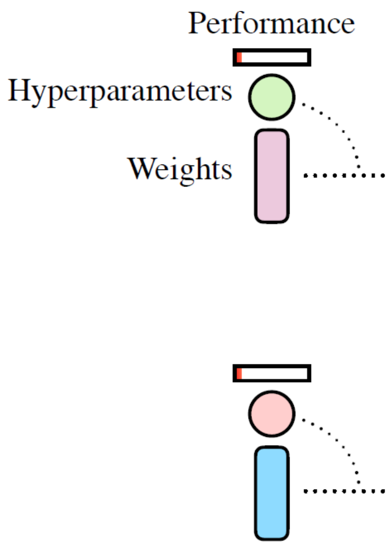
- *Population Based Training (PBT)*



- Start with random search
- Allow workers to **share information**
- Workers can **exploit** for model selection, and **explore** new hyperparameters
- Genetic algorithm acting on a timescale which allows gradient based learning

Proposed Method (Cont'd)

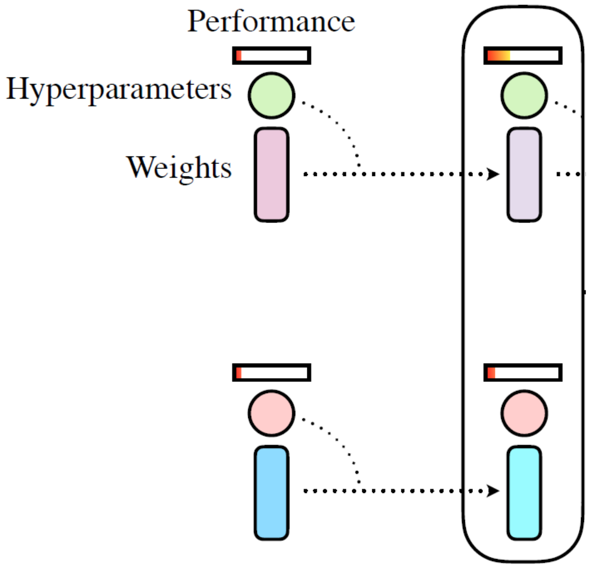
- *Population Based Training (PBT)*



- Randomly initialize model weights
- Randomly initialize hyperparameters from a prior distribution

Proposed Method (Cont'd)

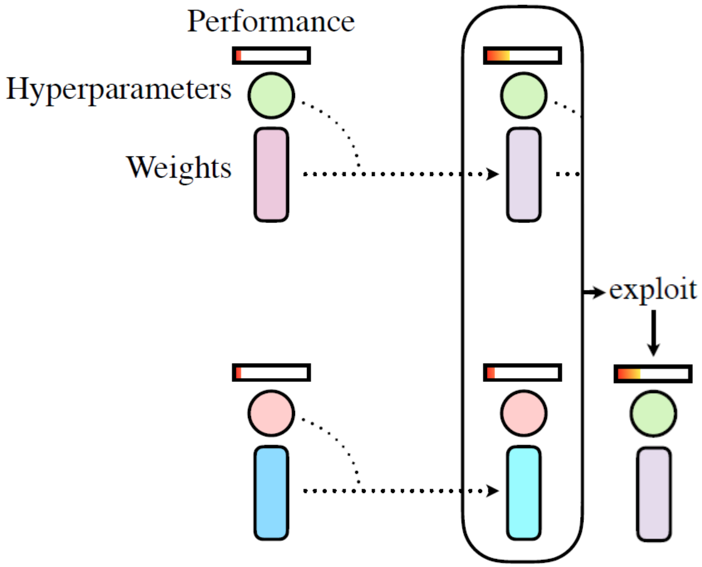
- *Population Based Training (PBT)*



  - Allow training for enough steps

Proposed Method (Cont'd)

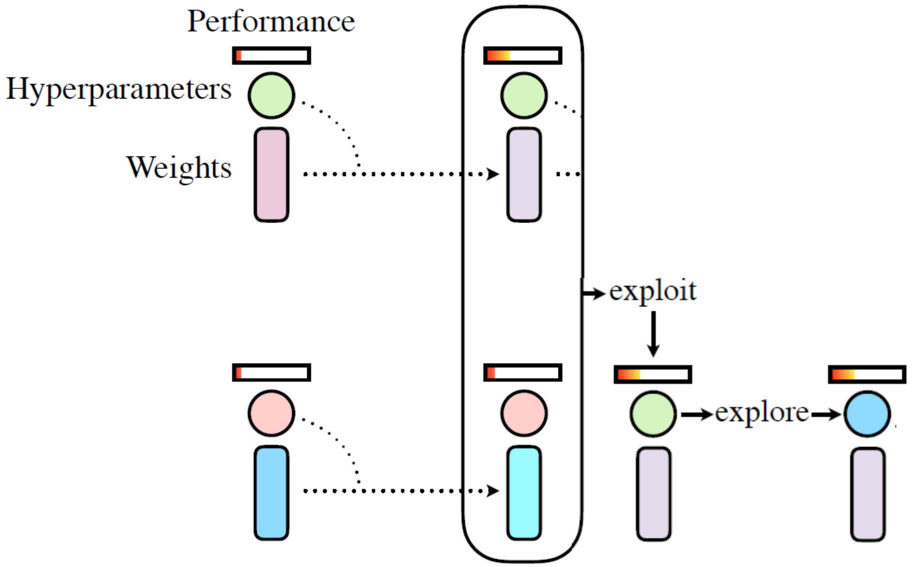
- *Population Based Training (PBT)*



- **Exploit**
  - Each workers compares its performance to the population. If bad, abandon it and replace the model and hyperparameter with better worker
  - **Binary tournament** – random two, better wins
  - **Truncation selection** – if in bottom 20%, replace with top 20%

Proposed Method (Cont'd)

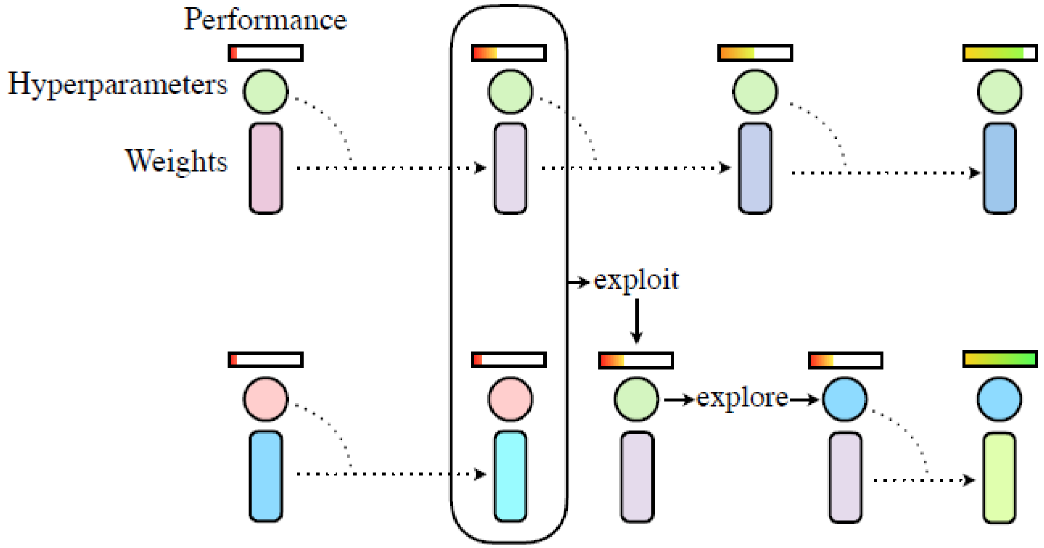
- *Population Based Training (PBT)*



- **Explore**
  - Mutate the hyperparameters that were replaced
  - **Perturb** – randomly perturbed by a factor e.g. 1.2
  - **Resample** – resampled from the initial prior distribution defined

Proposed Method (Cont'd)

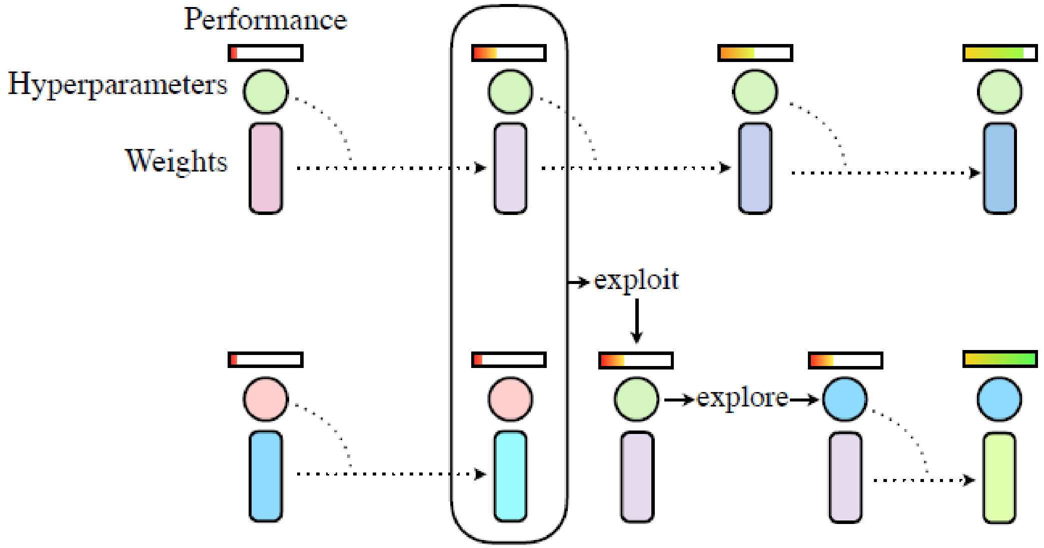
- *Population Based Training (PBT)*



- **Step:** perform steps of regular gradient-based training
- **Exploit:** if worker is bad, then replace it with better partial model
- **Explore:** mutate hyperparameters that replaced
- **Repeat**

Proposed Method (Cont'd)

- *Population Based Training (PBT)*



- Combines model optimization and hyperparameter refinement
- Exploit can optimize for non-differentiable & expensive metrices
  - accuracy on test set, BLEU scores, human normalized performance, …

- All workers benefit from the exploration luck

Experiments

- Apply PBT to 3 diff. learning problems
  - **RL** (Reinforcement Learning), **MT** (Machine Translation)
    GAN (Generative Adversarial Networks)

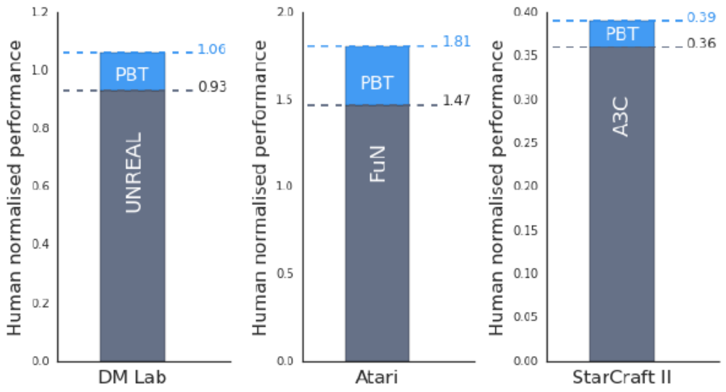
1. Deep Reinforcement Learning
  - Training of neural network agent with RL
    - Find a policy π to maximize expected episodic reward $E_\pi[R]$
  - 3 Tasks and models
    - DeepMind Lab, *UNREAL* (Jaderberg et al., 2016)
    - Atari games, *Feudal Networks* (Vezhnevets et al., 2017)
    - StarCraft 2, *A3C* baseline agents (Vinyals et al., 2017)

Experiments (Cont'd)

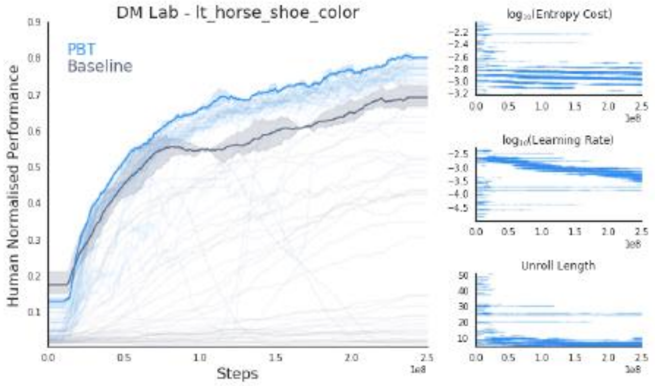
1. Deep Reinforcement Learning
   - Hyperparameters
     - learning rate, entropy cost, unroll length, intrinsic reward cost
   - Step
     - Step of gradient descent with RMSProp
   - Eval
     - Last 10 episodic rewards
   - Ready
     - between 10^6 ~ 10^7 agent steps have elapsed
   - Baseline
     - Random search with the same number of workers

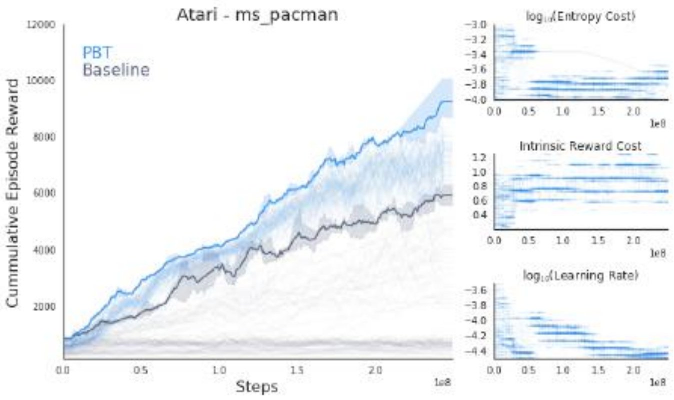
Experiments (Cont'd)

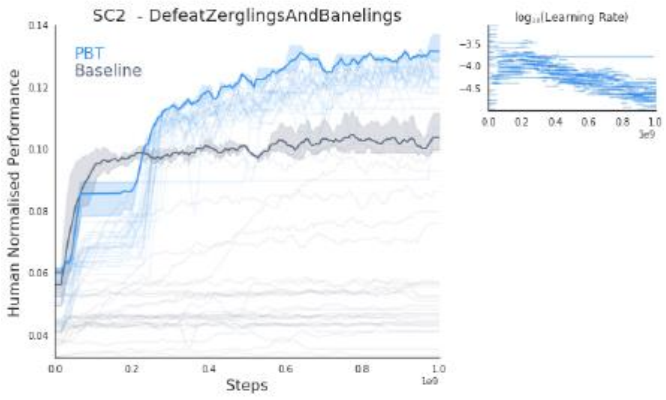
1. Deep Reinforcement Learning



workers : 40 80 30



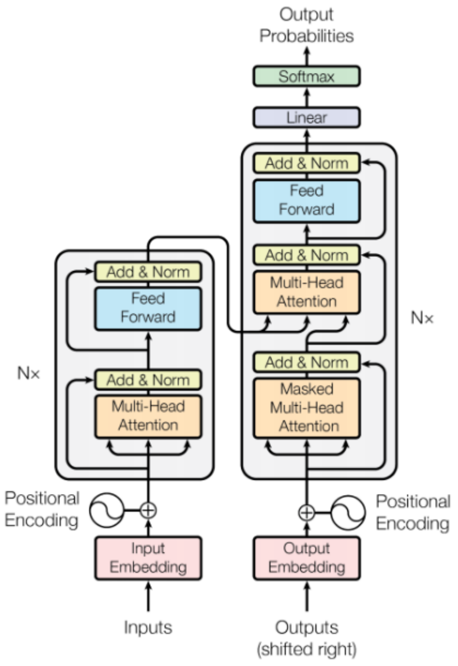




Experiments (Cont'd)

2. Machine Translation

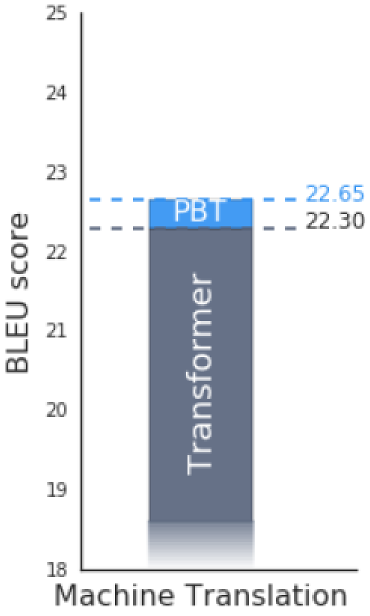
- *Transformer* networks (Vaswani et al., 2017), English to German on WMT 2014
- 32 workers with $400*10^3$ steps
- Hyperparameters
  - learning rate, attention dropout, layer dropout, ReLU dropout rates
- Step : step of Adam
- Eval : **BLEU score** on WMT *newstest2012* dataset
- Exploit : **binary tournament**
- Explore : **perturb** (1.2 or 0.8)

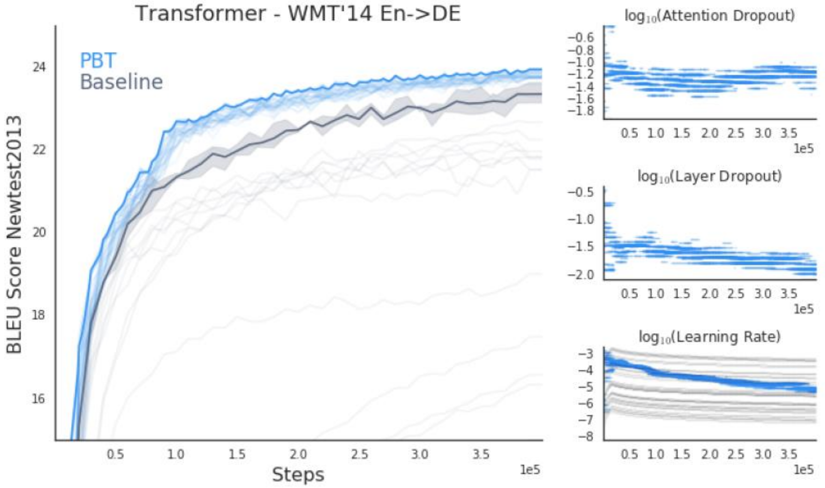


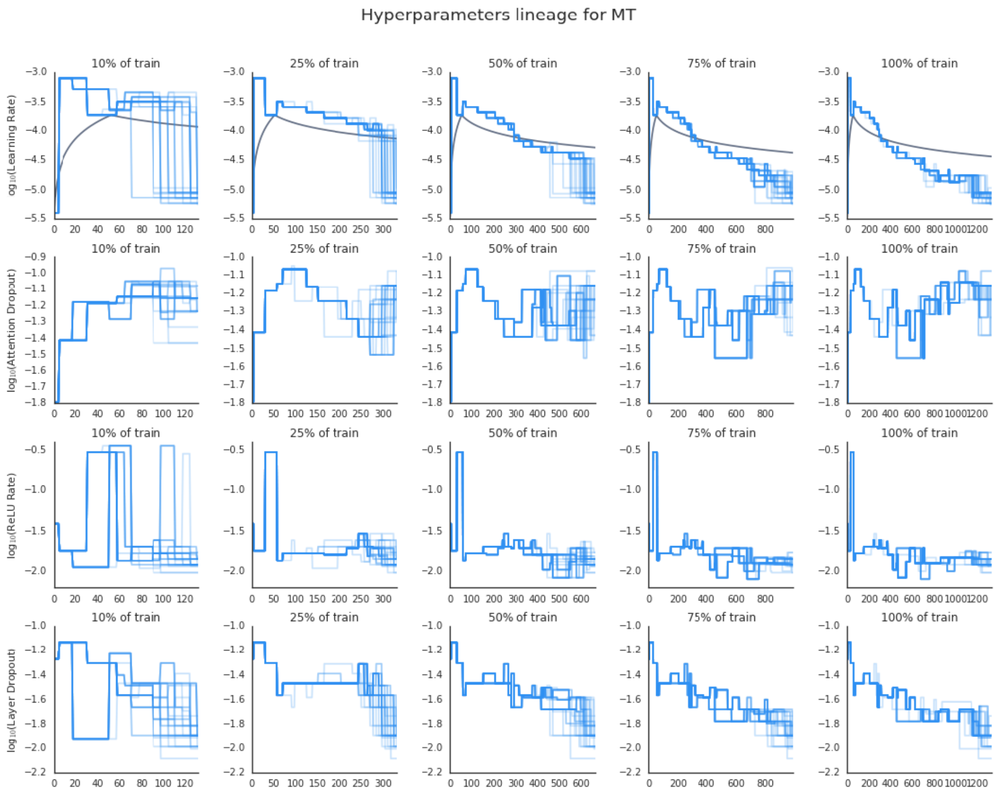
- Baseline model : highly optimized hyperparameter values
  - Result of hand tuning and Bayesian Optimization

Experiments (Cont'd)

2. Machine Translation



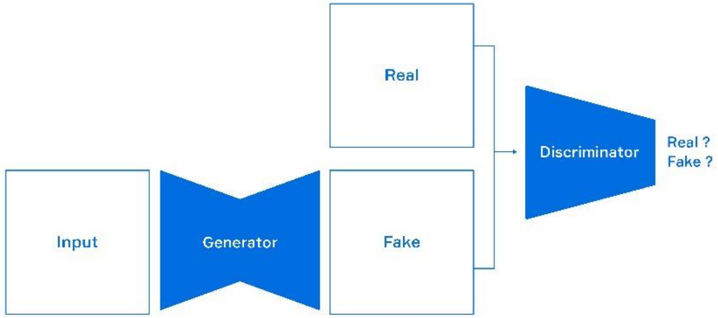




small Transformer network with small batch size

Experiments (Cont'd)

3. Generative Adversarial Networks



- Model : *DCGAN* (Radford et al., 2016) CIFAR trained
- Hyperparameters
  - G's learning rate, D's learning rate
- Eval
  - Inception score : $\mathrm{IS}(G) = \exp\left(\mathbb{E}_{\mathbf{x}\sim p_g} D_{KL}(p(y|\mathbf{x}) \parallel p(y)) \right)$
  - Outputs of pretrained CIFAR classifier

Good G
→ low of p(y|x), high of p(y)
→ high of IS(G)

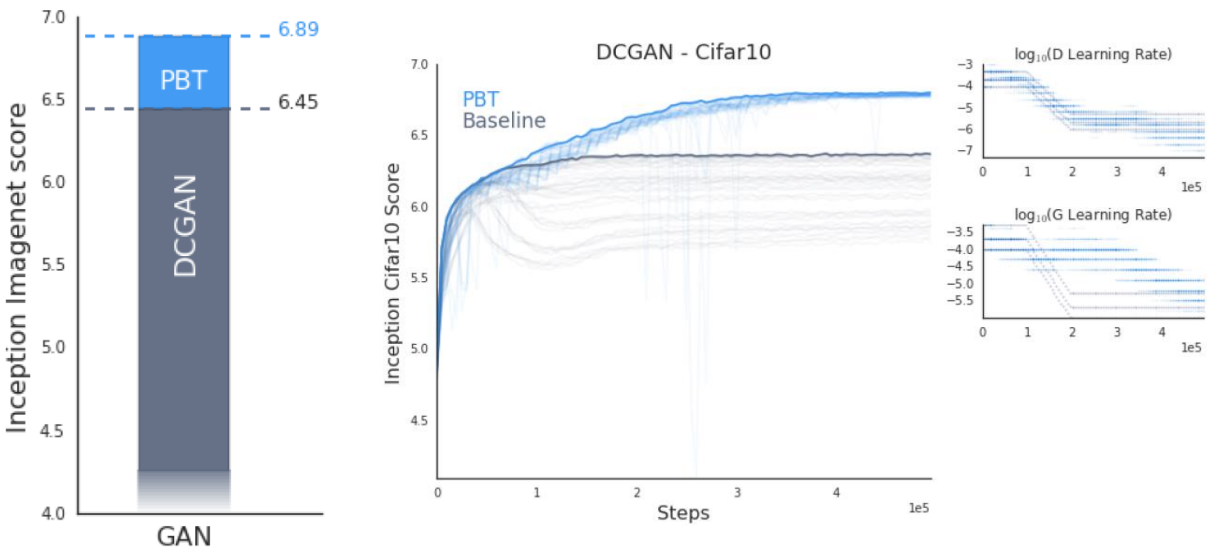
- Exploit : **both** (binary tournament, truncation selection)
- Explore : **perturb** (2.0 or 0.5)
- 45 workers

- Baseline model : Best among hand-design annealing strategies

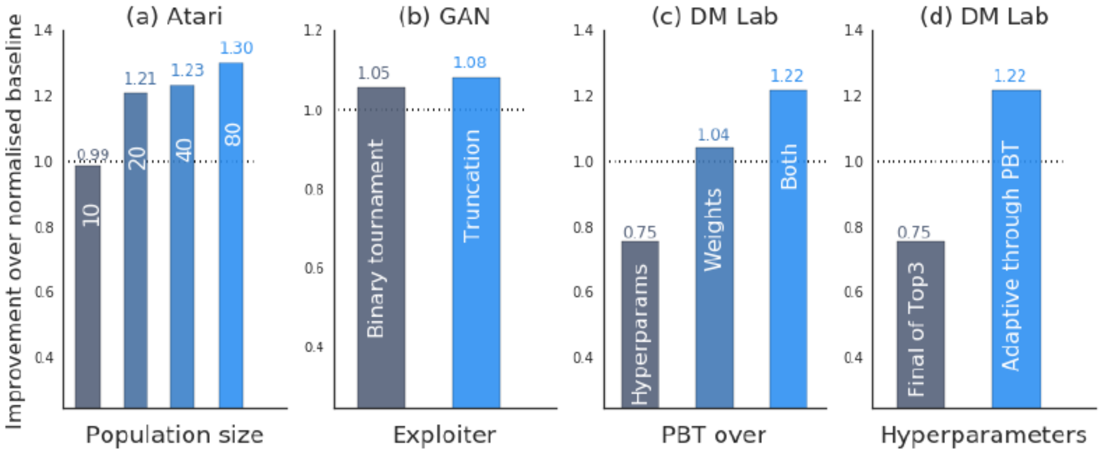
Experiments (Cont'd)

3. Generative Adversarial Networks

GAN results



Ablation study



Conclusion

- Proposed Population Based Training (PBT)
  - Based on Genetic algorithm
  - Bridges between **parallel search** and **sequential optimization** methods
  - Optimize over weights and hyperparameters jointly
  - Improvements in accuracy, stability across a wide range of domains
  - Discovers an **adaptive schedule** rather than fixed set of hyperparameters