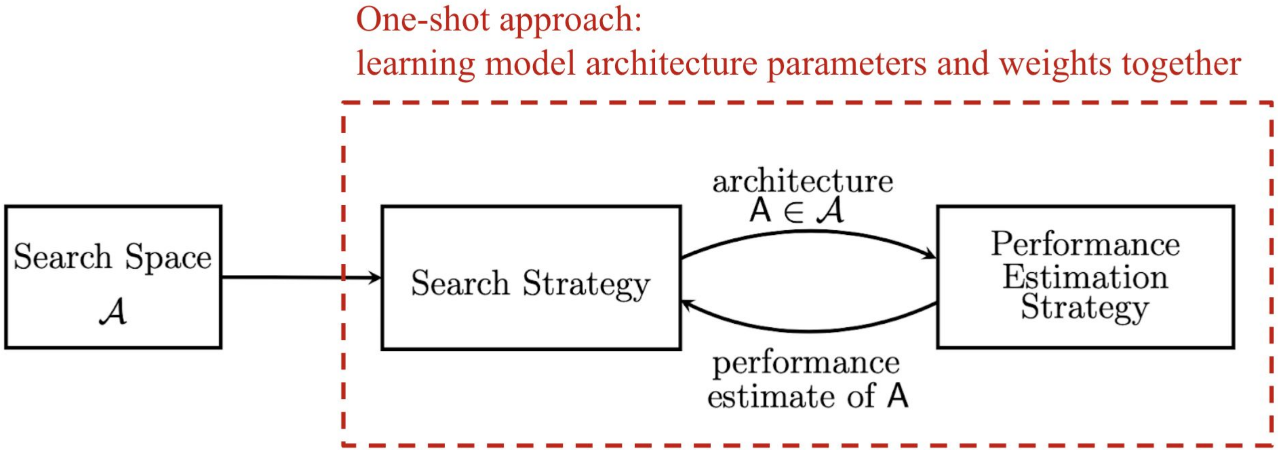
Training-Free NAS RNN

NAS: Problem statement


One-shot approach:
learning model architecture parameters and weights together
Search Space
$\mathcal{A}$
Search Strategy
architecture
$\mathsf{A} \in \mathcal{A}$
Performance Estimation Strategy
performance estimate of A

The idea of NAS is to automatically find an architecture which perform well on a certain task

Train-free NAS

While training-based NAS algorithms have achieved state-of-the-art performances in various tasks, their search costs are unaffordable in resource-constrained scenarios mainly due to their requirement for training DNNs during the search.

Idea:

Let’s find some heuristics to predict the final performance without training

(Pham et al., 2018; Liu et al., 2019)

ZeNAS

Motivation:
The expressivity of linear function class measured by Gaussian complexity is controlled by the Frobenius norm of its parameter matrix W

Idea:
Approximate this complexity using the following function.

$$Zen(f) = \log(\mathbb{E}_{\mathbf{x},\epsilon}\|f(\mathbf{x}) - f(\mathbf{x} + \alpha\epsilon)\|_F)$$

Link

TeNAS

Motivation:

Convergence in NTK case can be described using the following formula:

$$\mu_t\left(\boldsymbol{X}_{\text{train}}\right)_i = \left(\mathbf{I} - e^{-\eta\lambda_i t}\right)\boldsymbol{Y}_{\text{train},i}$$

Idea:

Introduce some scores based on NTK

$$Tenas(f) = \frac{\lambda_0}{\lambda_m},$$

$$\mathcal{M}_{\text{Trace}} \triangleq \sqrt{\left\|\Theta_0\right\|_{\text{tr}}/m}$$

GradScore

Simple metric can possibly approximate the final performance

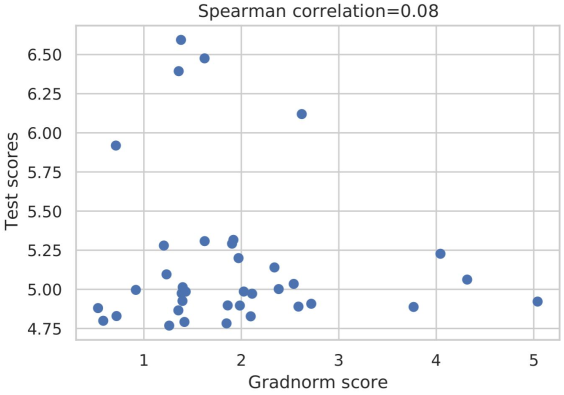
$$GradScore(f) = \mathbb{E}_{\mathbf{x}} \|\nabla f(\mathbf{x})\|_F$$

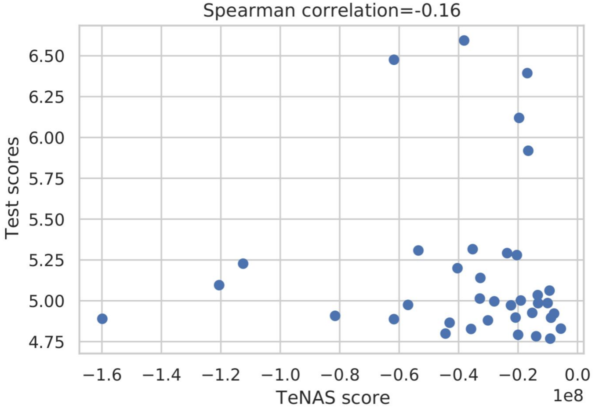
NAS-RNN

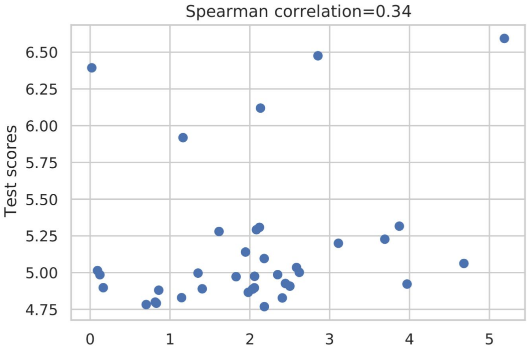
There are couple training-based NAS methods but there **no training methods.**

This work try to tackle this problem.

Heuristics performance







Thank you for attention!