**NEURAL ARCHITECTURE SEARCH**

**WITH REINFORCEMENT LEARNING**

**ABSTRACT**

1. RNN（recurrent neutral network） 循环神经网络
2. A validation set 验证集
3. From scratch 从头开始
4. Test set **accuracy** 测试集**精度**
5. Test set **perplexity** 测试集**困惑度**
6. **LSTM**(Long Short Term Memory) 长短期记忆递归 神经网络

**Target：**

maximize the expected accuracy of the generated architectures on a validation set.

**Methods:**

* Create a recurrent network (to generate the model descriptions of neural networks)
* Train this RNN with reinforcement learning

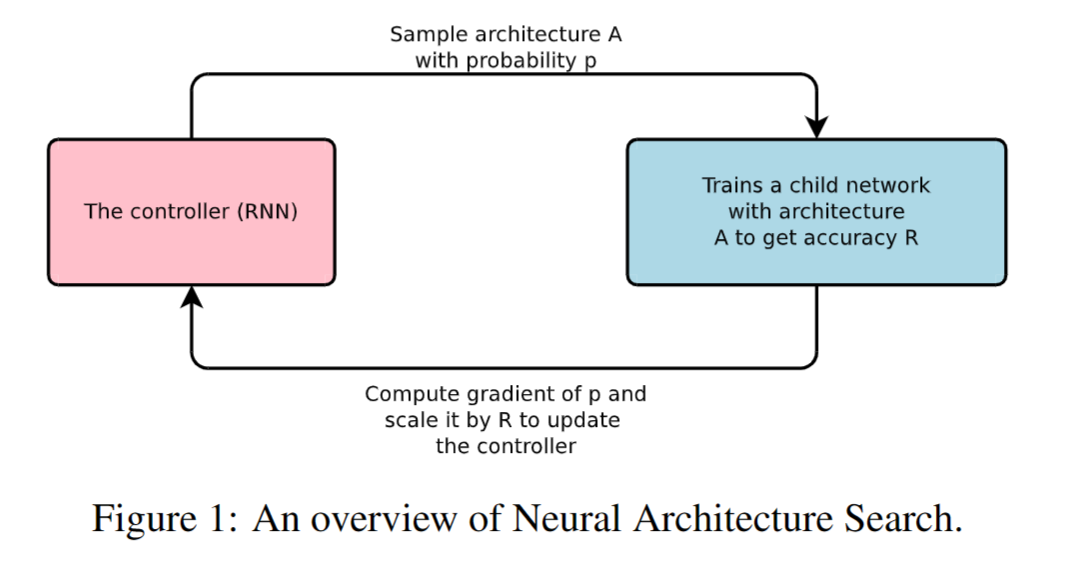
**Efforts**

* **CIFAR-10 model:**   
  0.09 percent better and 1.05x faster than the previous state-of-the-art model
* **Penn Treebank dataset model:**

3.6 perplexity better than the previous state-of-the-art mode

**1. INTRODUCTION**

1. Compute v. 计算
2. Iteration n, 迭代
3. Specified adj 特定的,指定的



**Type of NAS:** a gradient-based method for finding good architectures

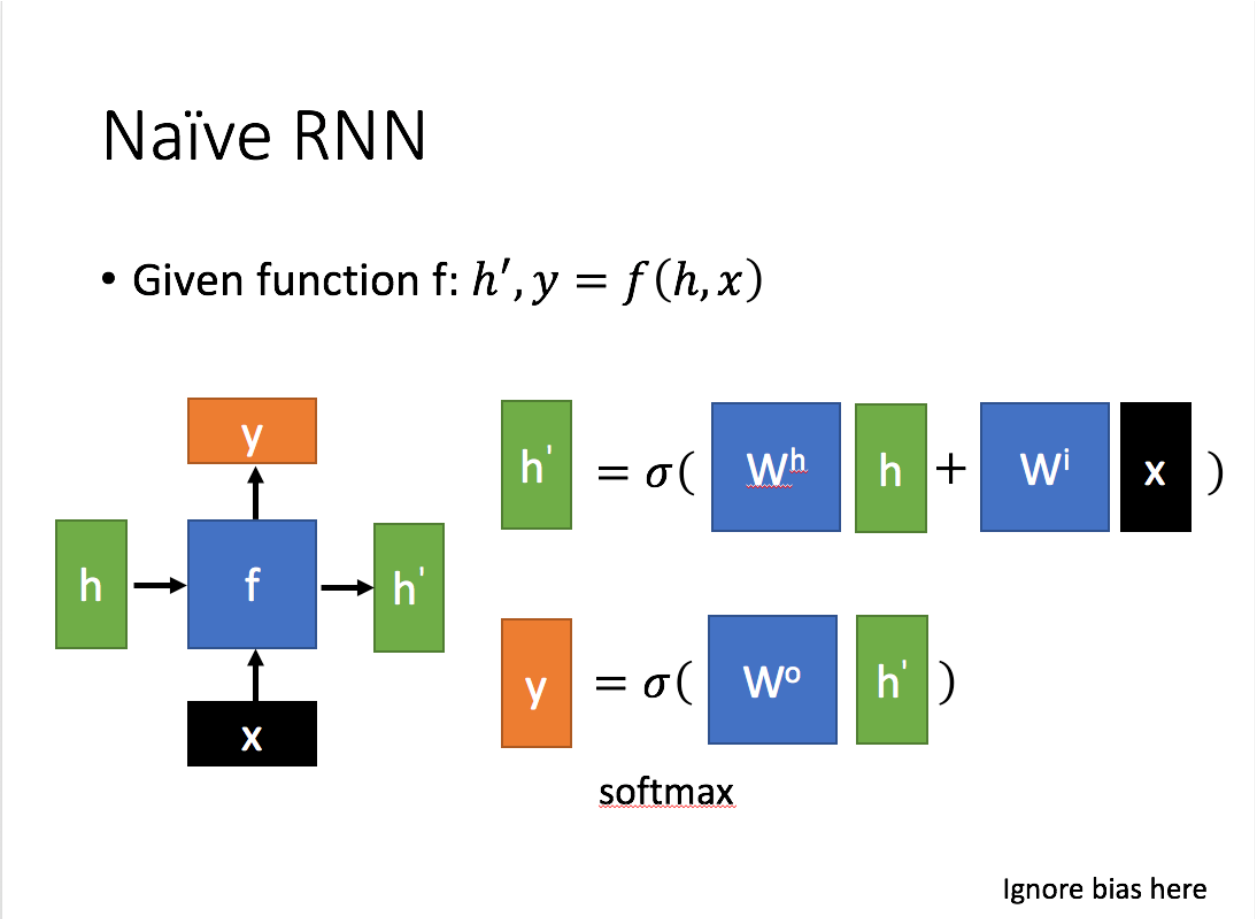
**BASE:** structure and connectivity of a neural network can be typically specified by a variable-length string.

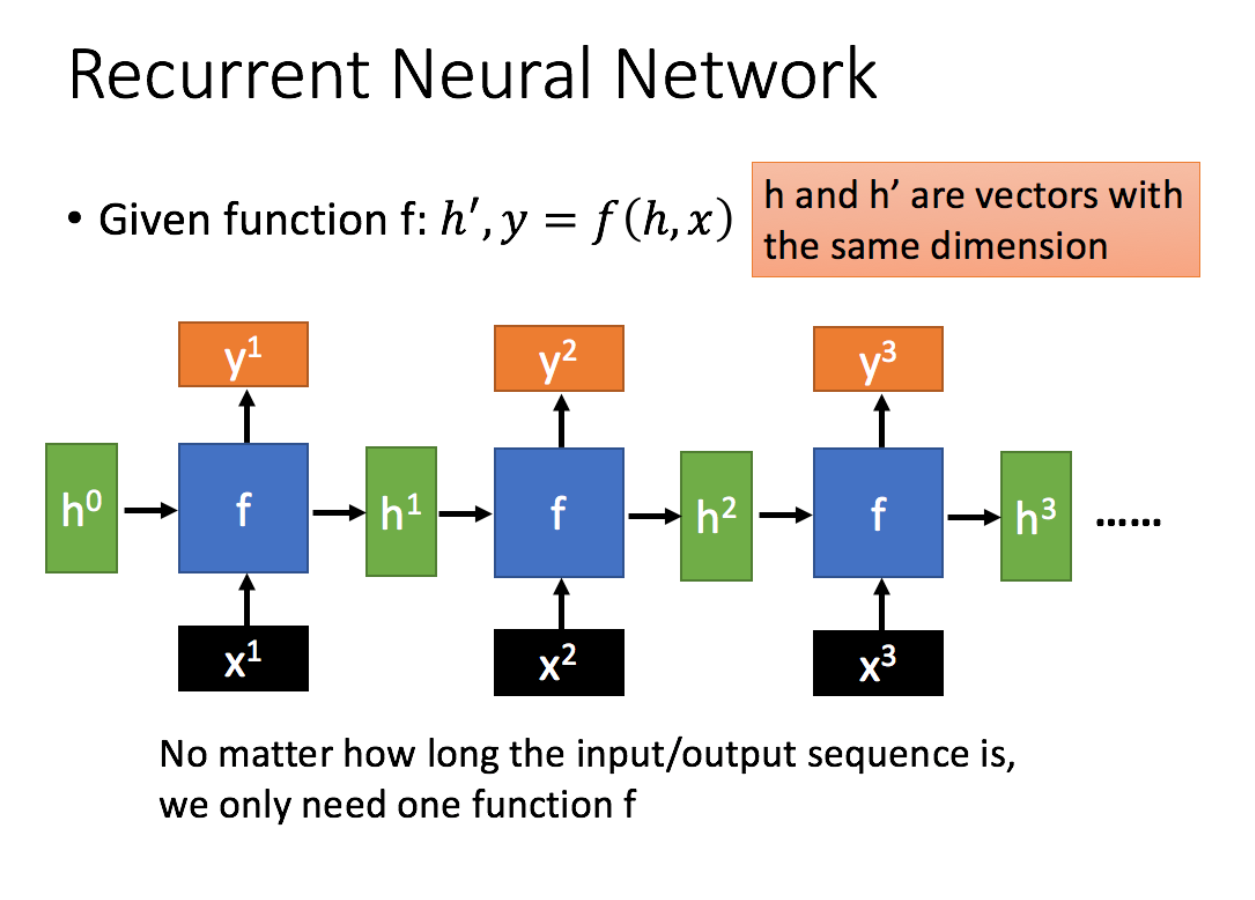
**2.RELATED WORK**

1. configuration n. 配置，设置

2. auto-regressive 自我回归

3. probabilistic program induction 概率程序规划





**3.METHODS**

1. skip connections 跳跃连接

2. sampled architectures 采样结构

3. implement v. 执行，实施

4. convergence n. 汇聚，聚集点

5. a held-out validation set 保留验证集

6.At convergence 在收敛时

7. held-out dataset 保留数据集

8. more concretely 更具体地说

9. non-differentiable adj. 不可微分的

10. variance n. 方差

11. unbiased gradient estimate 无偏梯度估计

12. empirical adj. 按照经验的，实证的

13. distributed training 分布式训练

14. replica n 复制品

15. attention mechanism 注意机制

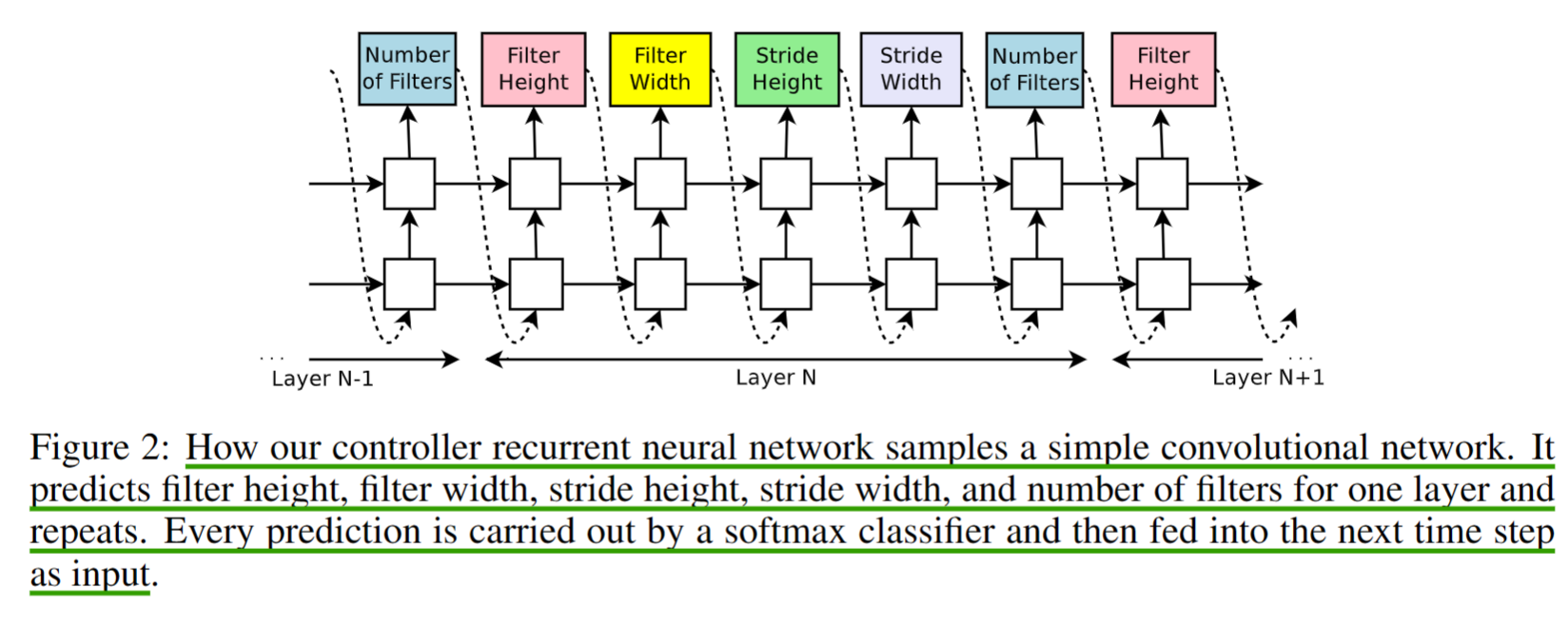
16. modifications n. 改正，修正

17. circumvent v. 规避，绕过

18.be compatible with 兼容于

**TARGET:** using a recurrent network to generate convolutional architectures

**STDEP1：Generate model descriptions with a controller recurrent neural network**



* RNN finish generating the architecture

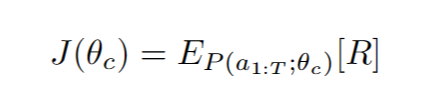
(the number of layers exceeds a certain value)

* the accuracy of the network on a held-out validation set is recorded
* the parameters of the controller RNN, θc, are optimized

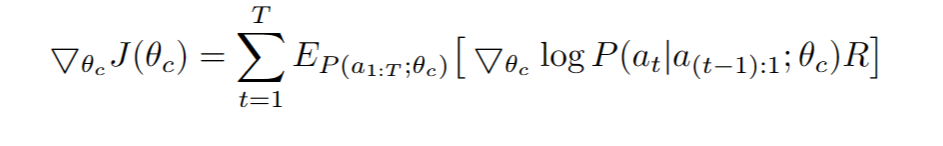
（maximize the expected validation accuracy of the proposed architectures）

**STDEP2：Training with reinforce**

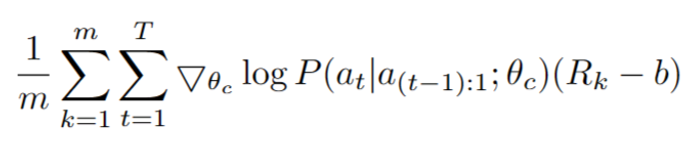
* find the optimal architecture, ask our controller to  
  maximize its expected reward, represented by J(θc)



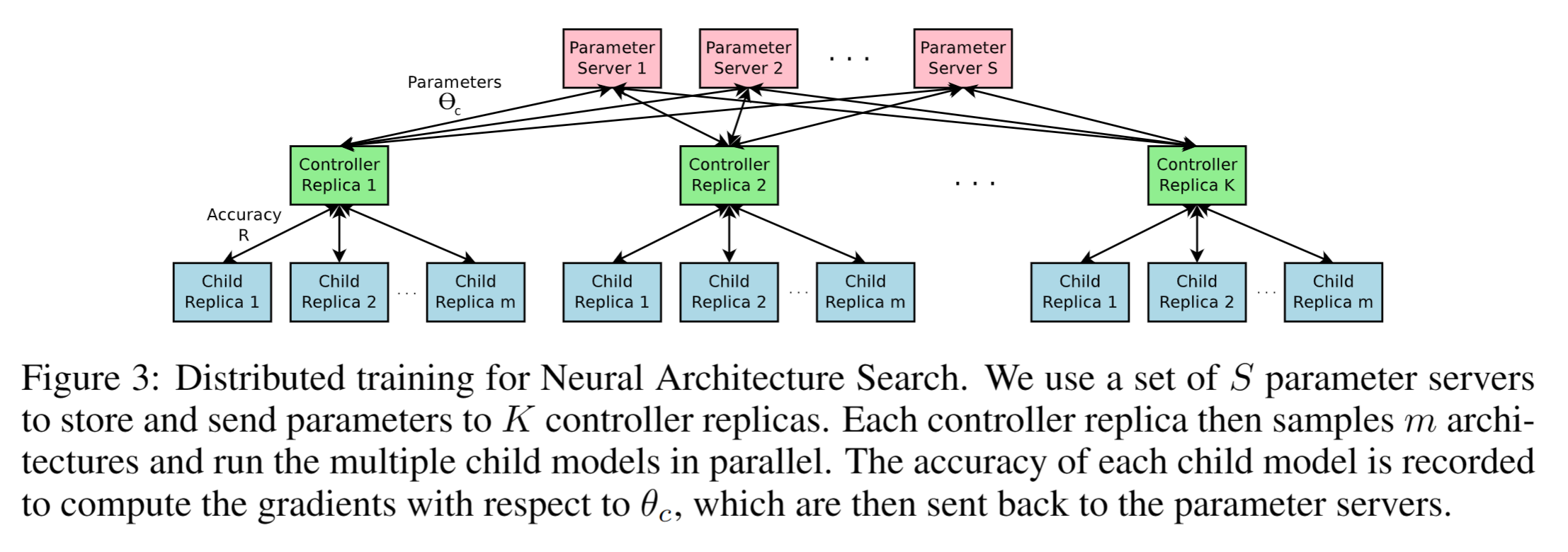
* use a policy gradient method to iteratively update θc by the reinforce rule.(when reward signal R is non-differentiable)



* reduce the variance of this estimate by employing a baseline function



**Accelerate Training with Parallelism and Asynchronous Updates**



* use **distributed training** and **asynchronous parameter updates**

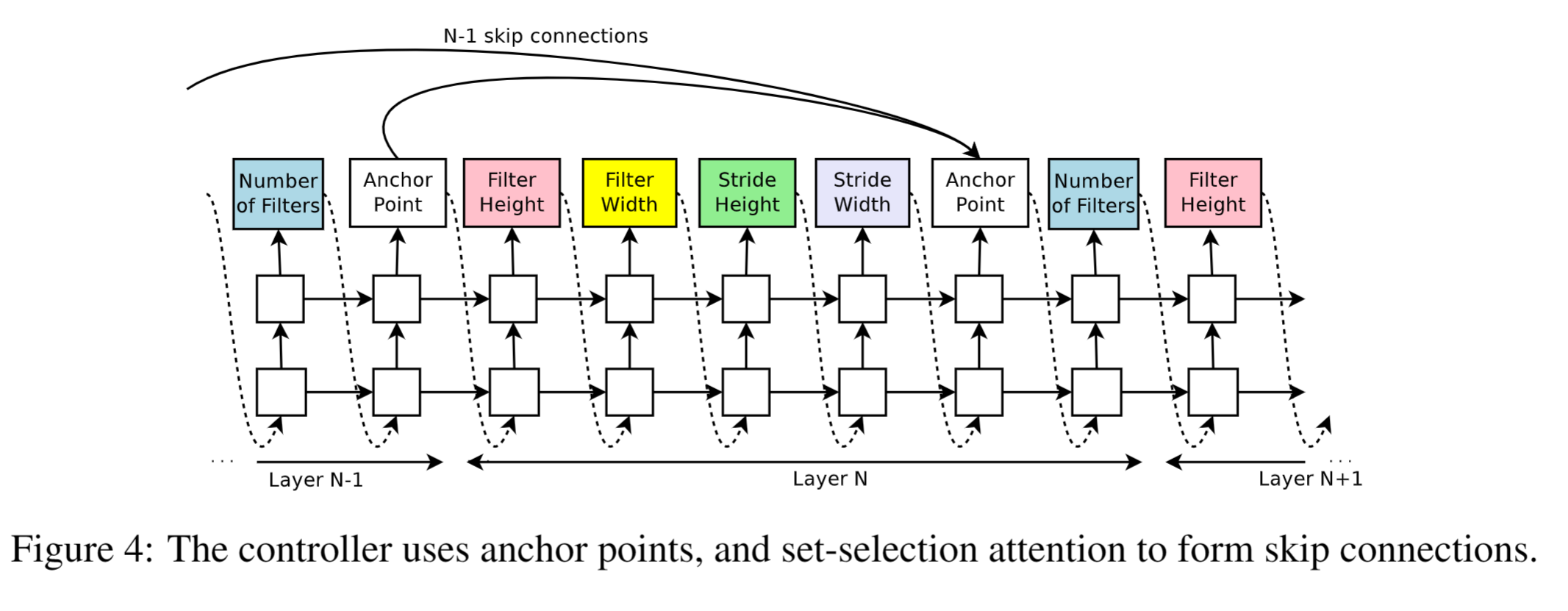
（In order to speed up the learning process of the controller)

* collects gradients according to the results of that minibatch of m architectures at convergence
* store the **shared parameters** for K controller replicas
* update the weights across all controller replicas
* each child network is reached

（when its training exceeds a certain number of epoch）

**STEP3：Increase architecture complexity with skip connections and other later types**

**TARGET：**allows our controller to propose skip connections or branching  
layers（widening the search space）



**METHOD：a set-selection type attention（**Attention mechanism）

**DISADVANTAGE:** cause **“compilation failures”**

**(**where one layer is not compatible with another layer, or one layer may not have any input or output.)

**CIRCUMVENT:**

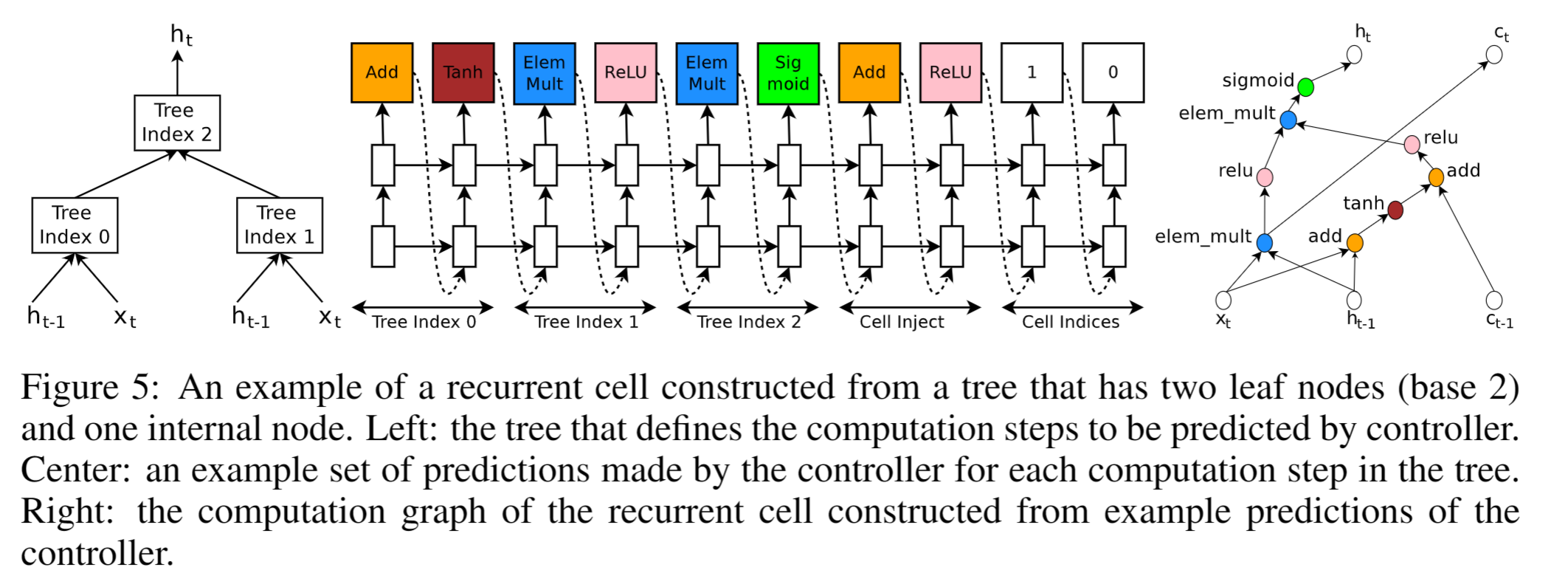
* the image is used as the input layer (If a layer is not connected to any input layer)
* at the final layer, we take all layer outputs that have not been connected and concatenate them before sending this final hidden state to the classifier.
* we pad the small layers with zeros so that the concatenated layers have the same sizes. (If input layers to be concatenated have different sizes)

**STEP4: Generate recurrent cell architectures**

**TARGET:** modify the above method to generate recurrent cells

**TYPE:** LSTM recurrent cell

**Base “2” architecture**



* each block specifying a combination method and an activation function for each tree index

1. **Experiments and results**
2. Excel v. 表现突出
3. Overfitting v. 过度拟合
4. Embedding n. 嵌入层

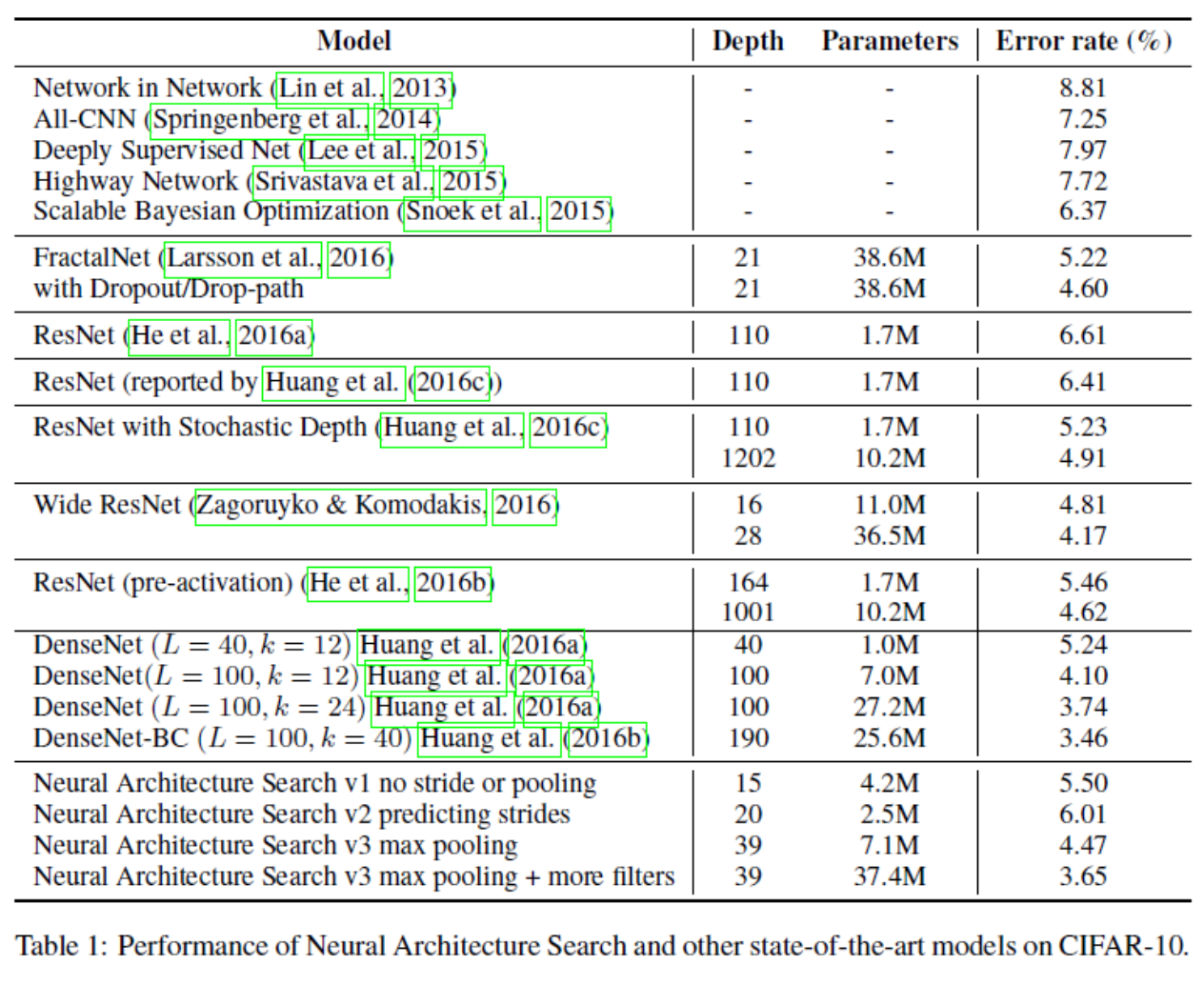
（[一文读懂Embedding的概念，以及它和深度学习的关系 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/164502624)

1. a grid search 网格搜索

**TARGET:**

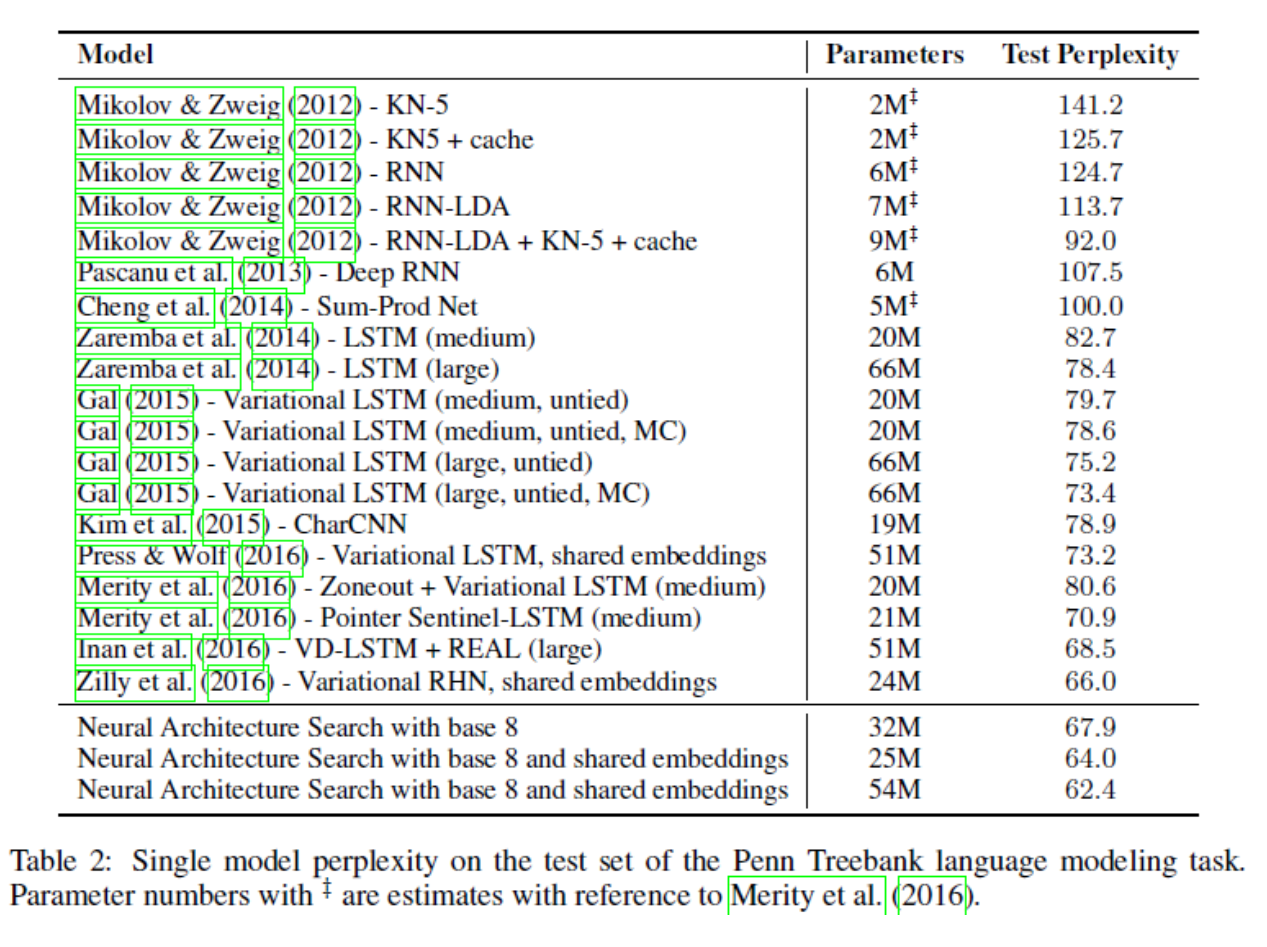
* find a good **convolutional architecture** on CIFAR-10
* find a good **recurrent cell** on Penn Treebank

**STEP1: Learning convolutional architectures for learning convolutional architectures for CIFAR-10**



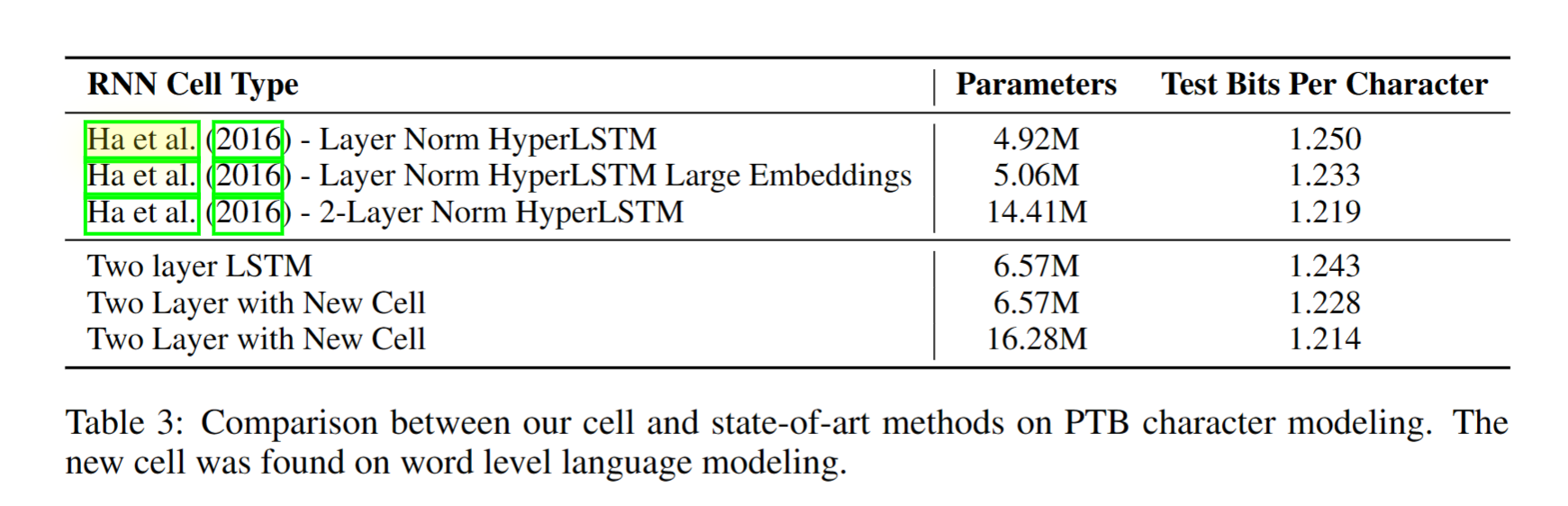
**RESULT**

**STEP2: Learning recurrent cells for Penn Treebank**

 **RESULT**

[炼丹术的终结——神经网络结构搜索之一 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/36301779)

**Transfer Learning Results:**



1. **CONCLUSION**

* method is flexible so that it can search variable-length architecture space by using recurrent network as the controller
* method has strong empirical performance on very challenging benchmarks and presents a new research direction for automatically finding good neural network architectures