Learning Architectures and Loss Functions in Continuous Space

Fei Tian
Machine Learning Group
Microsoft Research Asia

Self-Introduction

- Researcher @ MSRA Machine Learning Group
  - Joined in July, 2016
- Research Interests:
  - Machine Learning for NLP (especially NMT)
  - Automatic Machine Learning
- More Information: https://ustctf.github.io

Outline

- Overview
- Efficiently optimizing continuous decisions
  - *Loss Function Teaching*
- Continuous space for discrete decisions
  - *Neural Architecture Optimization*

Automatic Machine Learning

Automate every **decision** in machine learning

Learning rate,
Dropout,
Weight decay,
Temperature,
…

Architectures,
Depth,
Width,
Batch size,
…

Why Continuous Space?

- Life is easier if we have gradients
  - For example, we have a bunch of powerful gradient-based optimization algorithms

- Representation is compact
  - One of $|V|$ representations of words **V.S.** word embeddings

The Role of Continuous Space in AutoML

- For continuous decisions
  - How to efficiently optimize them?
    - And the more important, **elegantly**
- Our work:

Loss Function Teaching

- For discrete decisions
  - How to effectively cast them into continuous space?
- Our work:

Neural Architecture Optimization

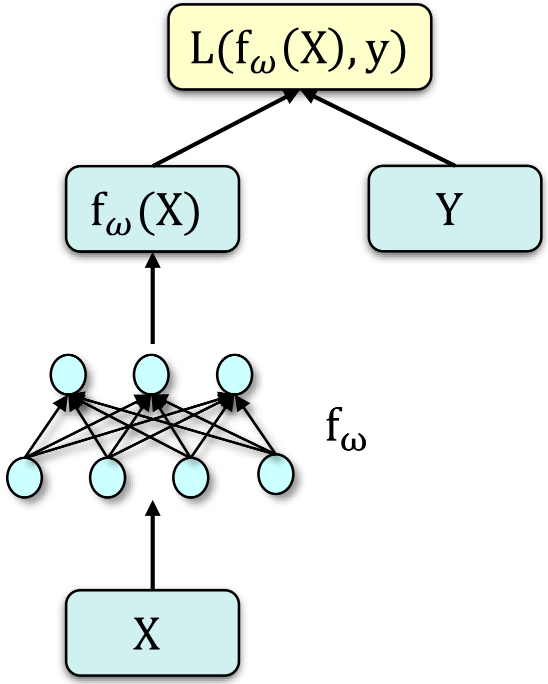
Learning to Teach with Dynamic Loss Functions

Lijun Wu, Fei Tian, Yingce Xia, Tao Qin, Tie-Yan Liu

NeurIPS 2018

Loss Function Teaching

- Recap to loss function $L(f_\omega(x), y)$
  - Typical examples:
    - Cross-Entropy: $L = -\log p(x) \cdot \vec{y}$, $\vec{y}_i = \mathbf{1}_{i=y}$
    - Maximum Margin: $L = \max_{y' \neq y} \log p_{y'} - \log p_y$
  - Learning objective of f_ω:
    - Minimize L
    - $\omega_t = \omega_{t-1} - \eta \frac{\partial L}{\partial \omega_{t-1}}$



- Objective of loss function teaching:

Discover best loss function L to train student model f_ω

  - Ultimate goal: improve the performance of f_ω

Why is it called “Teaching”?

- If we view model f_ω as *students*, then L is the *exams*

- Good teachers are **adaptive** :
  - They set good exams according to the status of the students

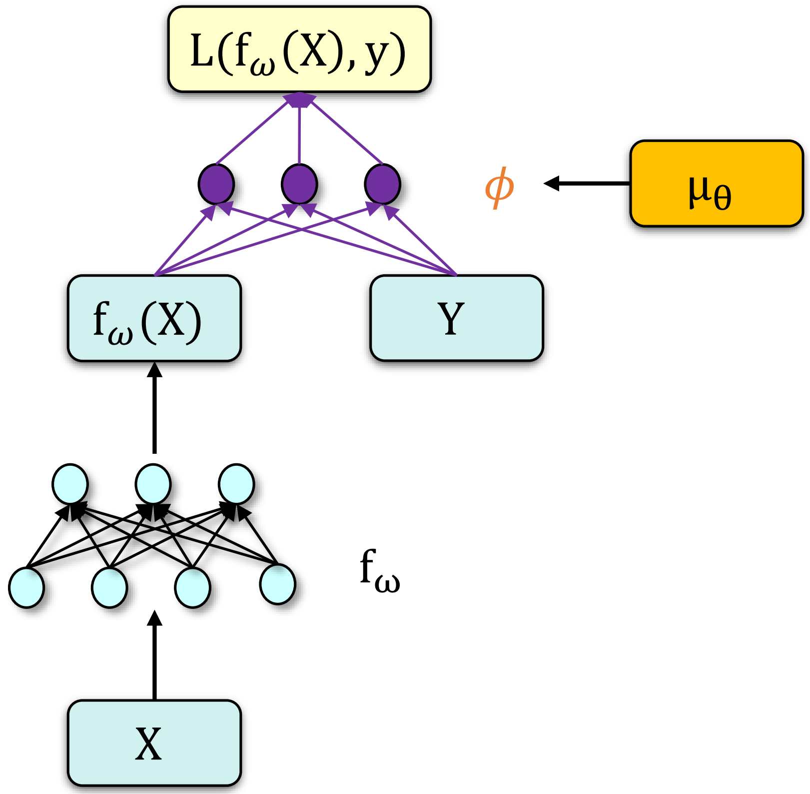
- An analogy:
  - *Data* (x, y) is the *textbook*
  - *Curriculum learning* schedules the textbooks (data) per the status of the student model

Can We Achieve Automatic Teaching?

- The first task: design a good decision space
- Our way: use another (parametric) neural network $L_{\phi}(f_{\omega}(x), y)$ as the loss function
  - The decision space: coefficients ϕ
  - It is **continuous**

Automatic Loss Function Teaching, cont.

- Assume the loss function itself is a neural network
  - $L_\phi(f_\omega(x), y)$, with ϕ as its coefficient
  - For example, generalized cross-entropy loss
    - $L_\phi = \sigma(-\log^T p(x)\, \mathrm{W}\vec{y} + b)$
    - $\phi = \{W, b\}$

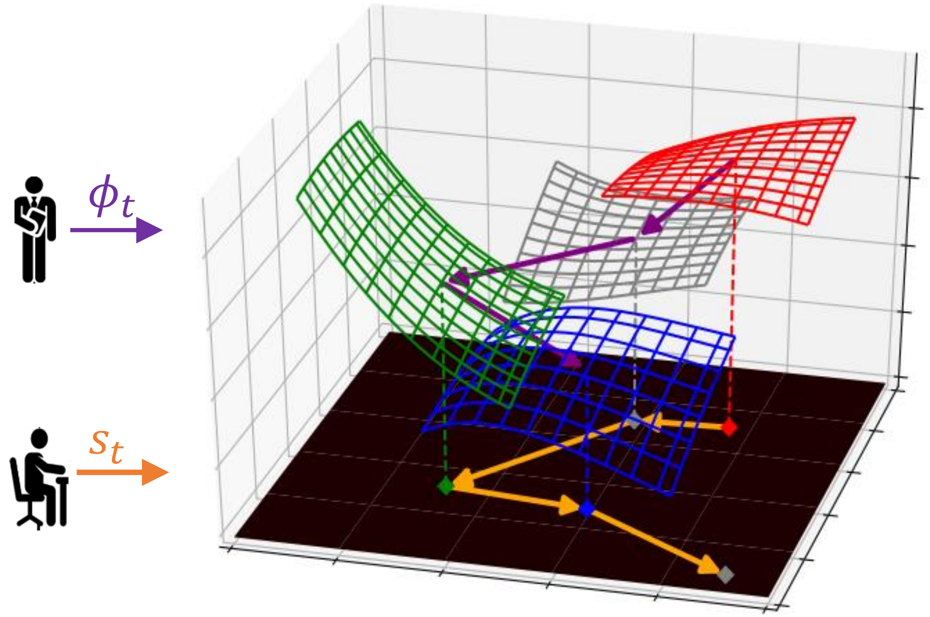


- A parametric teacher model μ_θ
  - Output ϕ
  - $\phi = \mu_\theta$

How to Be Adaptive?

- Extract feature s_t at different training step t of student model f_ω

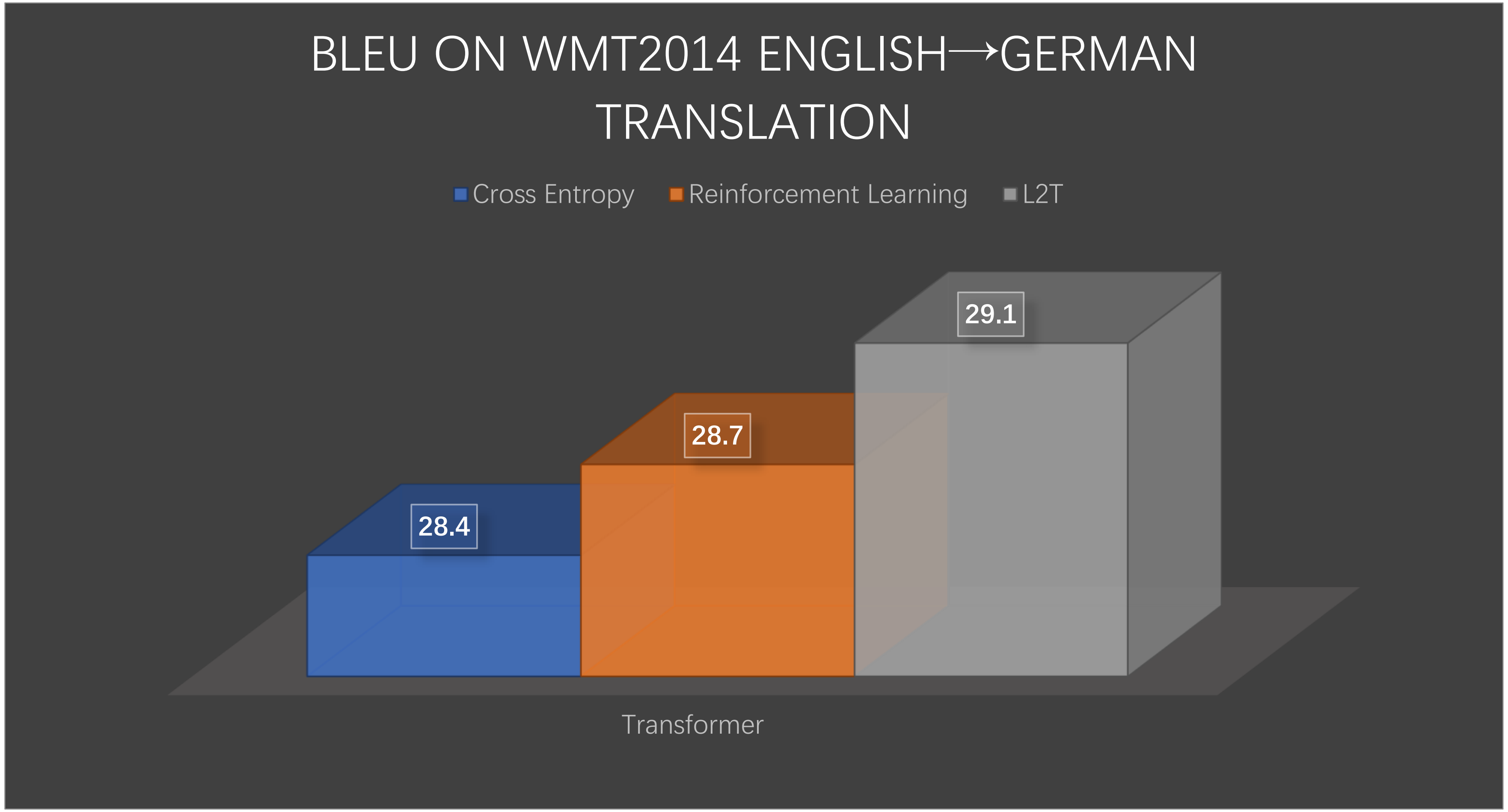
- The coefficients are adaptive
  - $\phi_t = \mu_\theta(s_t)$, generating adaptive loss functions $L_{\phi_t}(f_\omega(x), y)$



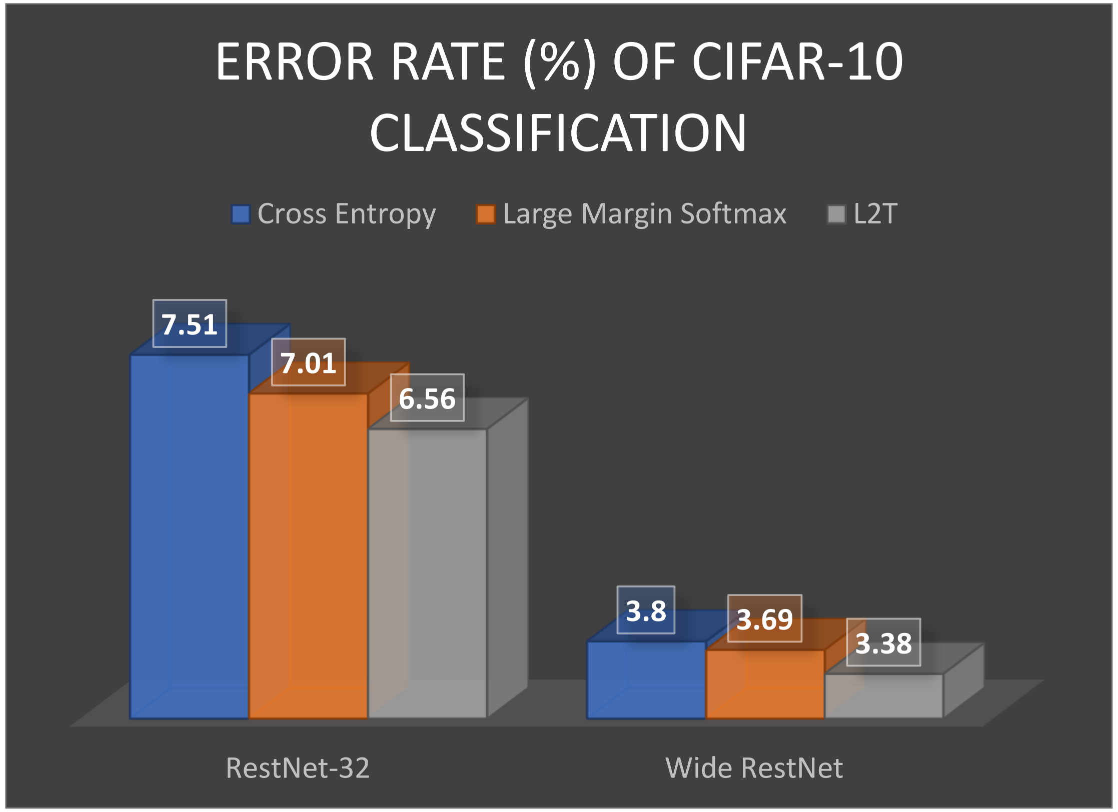
How to Optimize the Teacher Model?

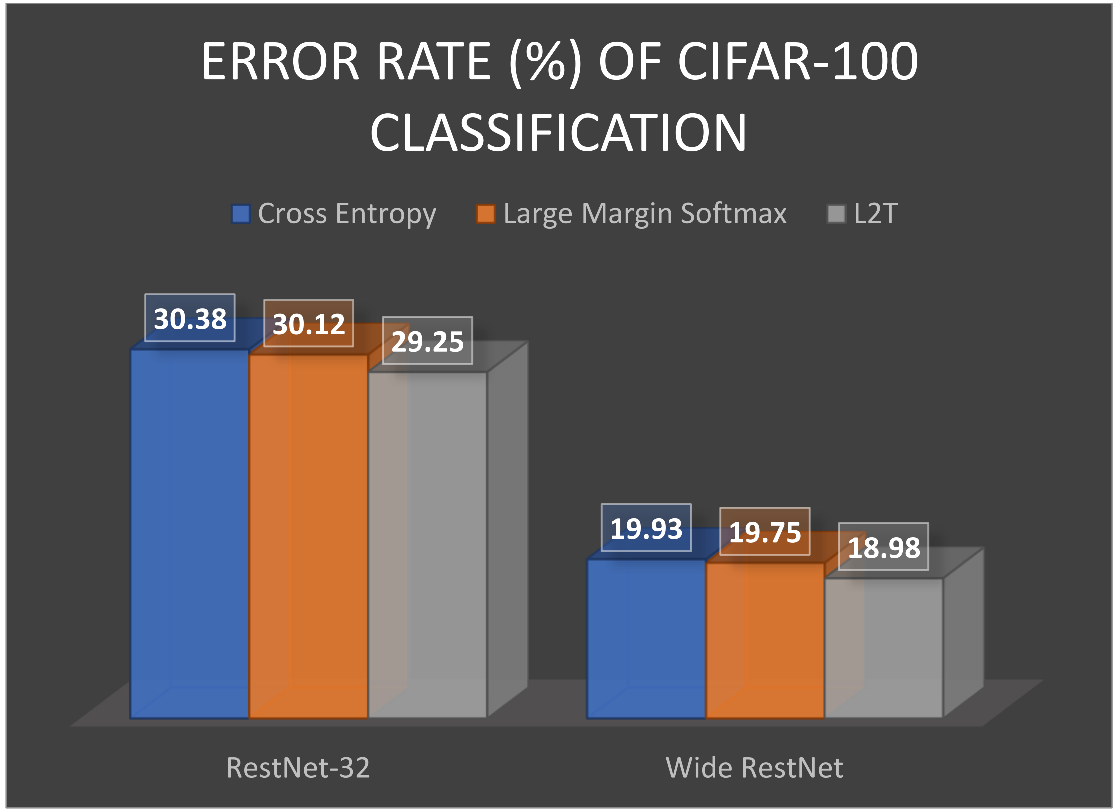
- Hyper gradient
- $\frac{\partial L_{dev}}{\partial \phi} = \frac{\partial L_{dev}}{\partial \omega_T} \frac{\partial \omega_T}{\partial \phi} = \frac{\partial L_{dev}}{\partial \omega_T} (\frac{\partial \omega_{T-t}}{\partial \phi} - \eta_{T-1} \frac{\partial^2 L_{train}(\omega_{T-1})}{\partial \omega_{T-1} \partial \phi})$

Neural Machine Translation Experiment



Experiments: Image Classification





Till now…

- We talked about how to set continuous decisions for a particular AutoML task
- And how to effectively optimize it
- But what would if the design space is **discrete**?

Neural Architecture Optimization

Renqian Luo, Fei Tian, Tao Qin, En-Hong Chen, Tie-Yan Liu

NeurIPS 2018

The Background: Neural Architecture Search

- There might be no particular need to introduce the basis…
- Two mainstream algorithms:
  - Reinforcement Learning and Evolutionary Computing

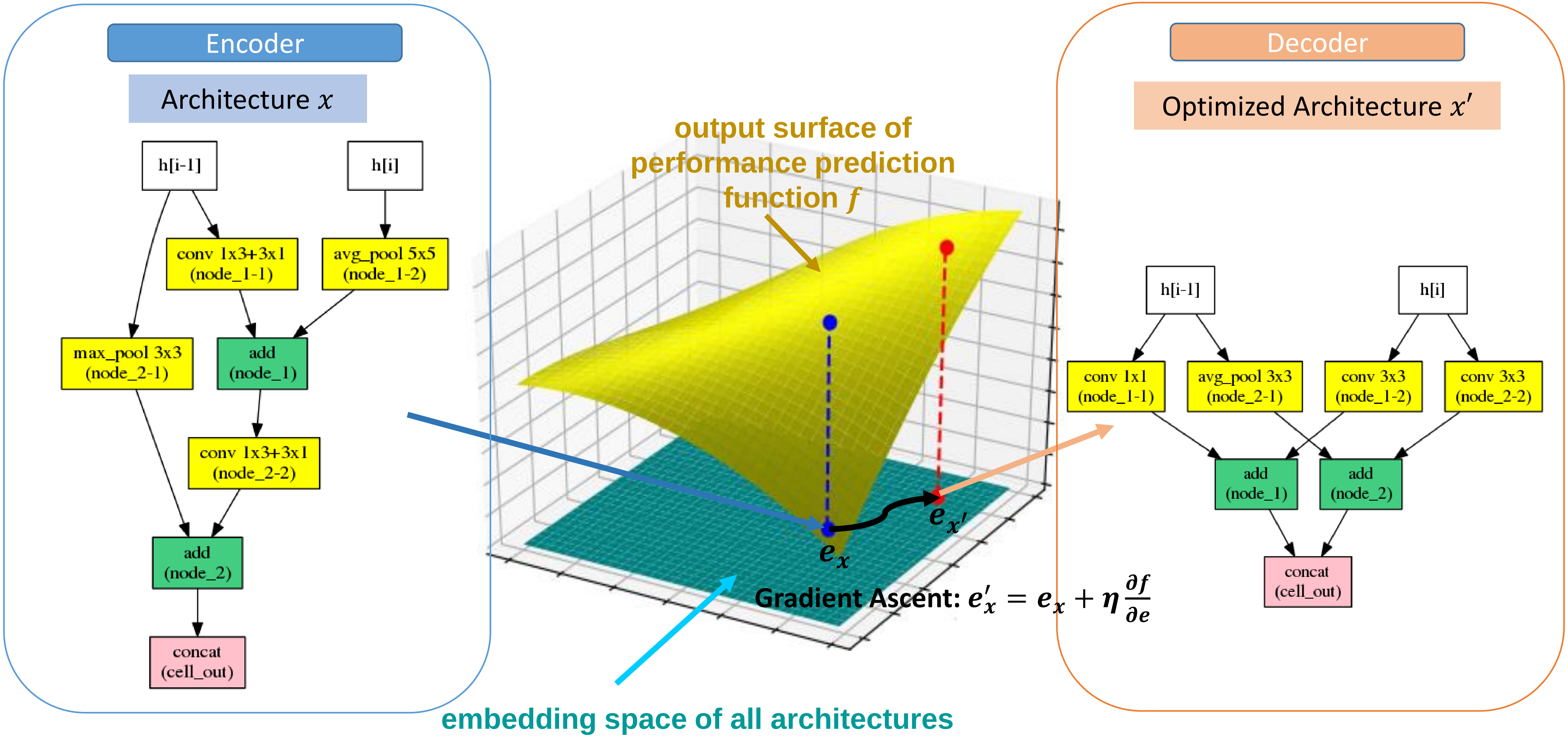
How to Cast the Problem into Continuous Space?

- Intuitive Idea

  Map the (discrete) architectures into continuous embeddings -> Optimize the embeddings -> Revert back to the architectures

- How to optimize?
  - Use the help of a performance predictor function f

How NAO Works?



Why the Encoder (including perf predictor) Could Work? Two Tricks

- Normalize the performance into $(0,1)$
  - Sometimes even with CDF

- Data augmentation
  - $(x, y) \rightarrow (x', y)$, if x and x' are symmetric
  - Improve the pairwise accuracy by 2% on CIFAR-10

Why the Decoder (i.e., perfect recovery) Could Work?

- Sentence-wise AutoEncoder with attention mechanism is easy to train
  - You can even obtain near 100 BLEU on test set!
  - So sometimes need *perturbations* to avoid trivial solution (e.g., in unsupervised machine translation [1,2])
  - f happens to be the *perturbation*

1. *Artetxe, Mikel, et al. "Unsupervised neural machine translation." ICLR 2018*
2. *Lample, Guillaume, et al. "Unsupervised machine translation using monolingual corpora only." ICLR 2018*

Experiments: CIFAR-10

Method	Error Rate	Resource (#GPU × #Hours)
ENAS	2.89	12
NAO-WS	2.80	7
AmoebaNet	*2.13*	*3150 * 24*
Hie-EA	*3.15*	*300 * 24*
NAO	*2.10*	*200 * 24*

Experiments: Transfer to CIFAR-100

Model	B	N	F	#op	Error (%)
DenseNet-BC [19]	/	100	40	/	17.18
Shake-shake [15]	/	/	/	/	15.85
Shake-shake + Cutout [12]	/	/	/	/	15.20
NASNet-A [48]	5	6	32	13	19.70
NASNet-A [48] + Cutout	5	6	32	13	16.58
NASNet-A [48] + Cutout	5	6	128	13	16.03
PNAS [27]	5	3	48	8	19.53
PNAS [27] + Cutout	5	3	48	8	17.63
PNAS [27] + Cutout	5	6	128	8	16.70
ENAS [37]	5	5	36	5	19.43
ENAS [37] + Cutout	5	5	36	5	17.27
ENAS [37] + Cutout	5	5	36	5	16.44
AmoebaNet-B [38]	5	6	128	19	17.66
AmoebaNet-B [38] + Cutout	5	6	128	19	15.80
NAONet + Cutout	5	6	36	11	15.67
NAONet + Cutout	5	6	128	11	14.36

Experiments: PTB Language Modelling

Method	Perplexity	Resource (#GPU × #Hours)
NASNet	62.4	1e4 CPU days
ENAS	58.6	12
NAO	56.0	300
NAO-WS	56.4	8

Experiments: Transfer to WikiText2

Models and Techniques	#params	Test Perplexity
Variational LSTM + weight tying [20]	28M	87.0
LSTM + continuos cache pointer [16]	-	68.9
LSTM [33]	33	66.0
4-layer LSTM + skip connection + averaged weight drop + weight penalty + weight tying [32]	24M	65.9
LSTM + averaged weight drop + Mixture of Softmax + weight penalty + weight tying [44]	33M	63.3
ENAS + weight tying + weight penalty [37] (searched on PTB)	33M	70.4
DARTS + weight tying + weight penalty (searched on PTB)	33M	66.9
NAO + weight tying + weight penalty (searched on PTB)	36M	66.5

Open Source

- https://github.com/renqianluo/NAO

Thanks!

We are hiring! Send me a message if you are interested:

fetia@microsoft.com

The Panel Discussion

- AutoML具体包括什么(网络结构搜索，超参数搜索，传统机器学习模型等)?
- AutoML与meta-learning的关系？
- NAS的局限性？如何完全除去人为干预？
- NAS与representation/transfer learning?
- 如何看待Random Search and Reproducibility for NAS
- RL or ES or SGD, gradient-based NAS是未来吗？