Learning Structured Representations

Artem Bochkarev

Moscow Institute of Physics and Technology

October 26, 2017

Plan

Motivation

Supervised Machine Learning Task

We have the dataset $\mathcal{D} = (\mathbf{X}, \mathbf{y}) = (\mathbf{x}_i, y_i)_{i=1}^m, \ \mathbf{x}_i \in \mathcal{X}, y_i \in \mathcal{Y}$.
Our goal is to find a function $f \in \mathcal{F}$, $f : \mathcal{X} \to \mathcal{Y}$ such that

$$f = \arg\min_{\mathcal{F}} L(f(\mathbf{X}), \mathbf{y}),$$

where L is a loss function (preferably differentiable).

Standard Setups

- Regression: $\mathcal{Y} = \mathbb{R}$
- Classification: $\mathcal{Y} = \{1, \dots, K\}$

Motivation

Problems

In many applications it is not clear how so state the problem as a classification of regression task.

- Image scene analysis
- Sentence parsing

Structured Prediction

In order to solve more complex tasks, we need to make space $\mathcal{Y}$ more complicated, for example graphs or even trees.

Motivation

Advantages

- If we are able to predict graph structures, this would solve very complex problems (many real-world structures can be represented with graphs)
- Potentially, it is possible to teach model that would make other models (the end of the mankind)

Issues

It is a non-trivial task to obtain key components in the problem statement:

- Approximation function f
- Loss function for scoring structured outputs

Plan

Problem Statement

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^m, \mathbf{x}_i \in \mathbb{R}^n, y_i \in \mathbb{R}$. Find approximation function $f : \mathbb{R}^n \to \mathbb{R}$ from model space $\mathcal{F}$, minimizing loss function L:

$$f^* = \arg\min_{f \in \mathcal{F}} L(f(\mathbf{X}), \mathbf{y})$$

$$L = \sqrt{\sum_{i=1}^{m} (y_i - f(\mathbf{x}_i))^2}$$

Symbolic Regression

Find all valid superpositions defined by grammar G:

$$B(g,g)|U(g)|S,$$

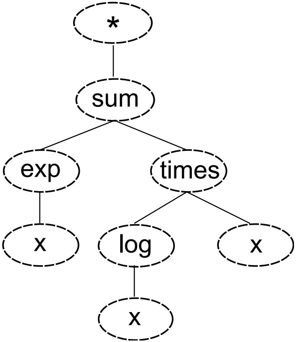
where B – binary operators, $\{+,-,*,/\}$, U – unary operators, $\{\ln, x^\alpha, \exp\}$, S – original variables.

Valid superpositions

1. elements are only generation functions g and original variables;
2. arity of element of superposition equals arity of used function;
3. the order of arguments corresponds to the order of arguments of used function;
4. domain of the next function is in the codomain of current function.

Tree of a superposition

Each superposition f corresponds to the tree of superposition Γ_f.
Depth of a superposition is a depth of the corresponding tree.



$$f = e^x + x \cdot (\log x)$$

Tree Γ_f

1. Root - $*$;
2. $V_i \mapsto g_r$;
3. val$(V_j) = v(g_{r(i)})$;
4. dom$(g_{r(i)}) \supset$ cod$(g_{r(j)})$;
5. arguments g_r are ordered;
6. x_i — leaves Γ_f.

Genetic algorithm

Generating superpositions with genetic algorithm

1: **while** required accuracy is not achieved **do**
2: Select subset of models, which minimizes loss function L, from population $\mathcal{M}$
3: Swap subtrees of two random models to obtain new valid superposition (permutation)
4: Replace random subtree with a new random one (mutation)
5: Add newly generated models to the population $\mathcal{M}$.
6: **end while**

Kulunchakov, A. S., V. V. Strijov. Generation of simple structured information retrieval functions by genetic algorithm without stagnation. *Expert Systems with Applications* 85 (2017): 221-230.

Plan

Problem

Approach

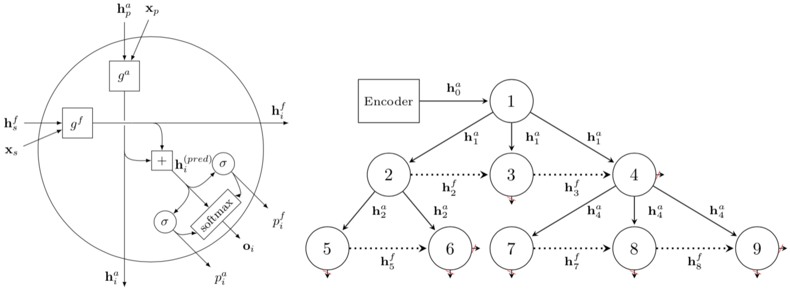
Reconstruct trees using encoder-decoder framework. This paper focuses on decoding trees from latent representations.

Architecture

Top-down, recursive, using doubly-recurrent neural network. Both the ancestral (parent-to-children) and the fraternal (sibling-to-sibling) flows of information are modeled with recurrent modules.

Alvarez-Melis, D., Jaakkola, T. S. (2017). Tree-structured decoding with doubly-recurrent neural networks.

Model architecture



Alvarez-Melis, D., Jaakkola, T. S. (2017). Tree-structured decoding with doubly-recurrent neural networks.

Model structure

Definitions

Let $\mathcal{T} = \{\mathcal{V}, \mathcal{E}, \mathcal{X}\}$, be an undirected labeled tree.

- $\mathcal{V}$ are vertices
- $\mathcal{E}$ are edges
- $\mathcal{X}$ are vertex labels

For a node $i \in \mathcal{V}$ denote parent as $p(i)$ and previous sibling as $s(i)$. Let g^a and g^f be functions which apply one step of the two separate RNNs.

Alvarez-Melis, D., Jaakkola, T. S. (2017). Tree-structured decoding with doubly-recurrent neural networks.

Model Structure

Hidden states update

$$\mathbf{h}_i^a = g^a(\mathbf{h}_{p(i)}^a, \mathbf{x}_{p(i)})$$

$$\mathbf{h}_i^f = g^f(\mathbf{h}_{s(i)}^f, \mathbf{x}_{s(i)})$$

Predictive hidden state

$$\mathbf{h}_i^{(pred)} = \tanh(\mathbf{U}^f \mathbf{h}_i^f + \mathbf{U}^a \mathbf{h}_i^a),$$

where $\mathbf{U}^f \in \mathbb{R}^{n \times D_f}$ and $\mathbf{U}^a \in \mathbb{R}^{n \times D_a}$ are learnable parameters. This state is used to predict a label for a node.

Alvarez-Melis, D., Jaakkola, T. S. (2017). Tree-structured decoding with doubly-recurrent neural networks.

Node prediction

Topological probabilities

$$p_i^a = \sigma(\mathbf{u}^a \cdot \mathbf{h}_i^{(pred)})$$
$$p_i^f = \sigma(\mathbf{u}^f \cdot \mathbf{h}_i^{(pred)})$$

Label prediction

$$\mathbf{o}_i = \mathsf{softmax}(\mathbf{W}\mathbf{h}_i^{(pred)} + \alpha_i \mathbf{v}^a + \varphi_i \mathbf{v}^f),$$

where $\alpha_i, \varphi_i \in \{0, 1\}$ are binary variables indicating the topological decisions.

Alvarez-Melis, D., Jaakkola, T. S. (2017). Tree-structured decoding with doubly-recurrent neural networks.

Forward pass

Generation procedure

After the node's output symbol $\mathbf{x}_i$ has been obtained by sampling from $\mathbf{o}_i$, the cell passes $\mathbf{h}_i^a$ to all its children and $\mathbf{h}_i^f$ to the next sibling (if any), enabling them to realize their states.
This procedure continues recursively, until termination conditions cause it to halt.

Loss function

$$\mathcal{L}(\hat{\mathbf{x}}) = \sum_{i \in \mathcal{V}} \mathcal{L}^{label}(\mathbf{x}_i, \hat{\mathbf{x}}_i) + \mathcal{L}^{topo}(\mathbf{p}_i, \hat{\mathbf{p}}_i),$$

the former is a cross-entropy loss, the latter is a binary cross-entropy loss.

Alvarez-Melis, D., Jaakkola, T. S. (2017). Tree-structured decoding with doubly-recurrent neural networks.

Backward pass

Gradient computation

1. Gradient of the current node's label prediction loss w.r.t. softmax layer parameters $\mathbf{W}, \mathbf{v}^a, \mathbf{v}^f$: $\nabla_\theta \mathcal{L}(\mathbf{x}_i, \hat{\mathbf{x}}_i)$
2. Gradients of topological prediction variables loss with respect to sigmoid layer parameters: $\nabla_\theta \mathcal{L}(p_i^a, t_i^a)$ and $\nabla_\theta \mathcal{L}(p_i^f, t_i^f)$
3. Gradient of predictive state parameters with respect to $\mathbf{h}^{(pred)}$
4. Gradient of predicted ancestral and fraternal hidden states with respect to g^f and g^a's parameters.

Alvarez-Melis, D., Jaakkola, T. S. (2017). Tree-structured decoding with doubly-recurrent neural networks.

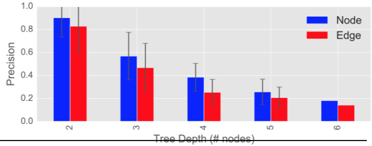
Experiment 1

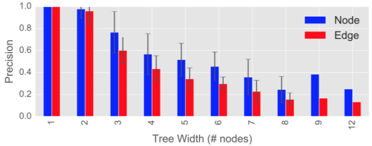
Problem

Synthetic dataset of randomly generated trees with English letters as node labels.

Evaluation loss

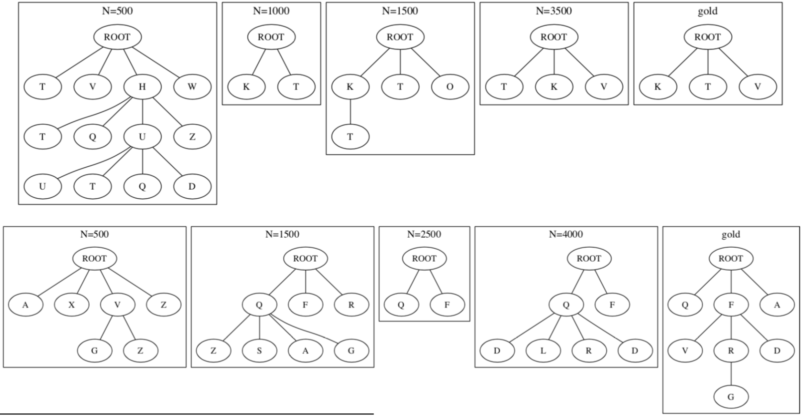
Precision and recall of recovering nodes and edges present in the gold tree.





Alvarez-Melis, D., Jaakkola, T. S. (2017). Tree-structured decoding with doubly-recurrent neural networks.

Experiment 1

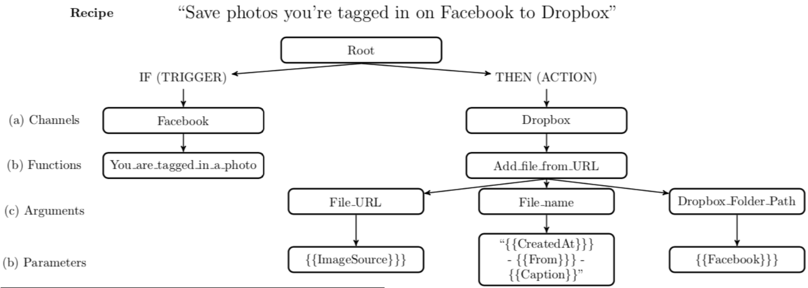


Alvarez-Melis, D., Jaakkola, T. S. (2017). Tree-structured decoding with doubly-recurrent neural networks.

Experiment 2

Problem

IFTTT (IF This Then That) dataset. The goal is to parse natural language sentence to tree recipe representation.



Alvarez-Melis, D., Jaakkola, T. S. (2017). Tree-structured decoding with doubly-recurrent neural networks.

Experiment 2

Method	Channel	+Func	F1
retrieval	36.8	25.4	49.0
phrasal	27.8	16.4	39.9
sync	26.7	15.4	37.6
classifier	64.8	47.2	56.5
posclass	67.2	50.4	57.7
SEQ2SEQ	68.8	50.5	60.3
SEQ2TREE	69.6	51.4	60.4
GRU-DRNN	70.1	51.2	62.7
LSTM-DRNN	**74.9**	**54.3**	**65.2**

Method	Channel	+Func	F1
retrieval	43.3	32.3	56.2
phrasal	37.2	23.5	45.5
sync	36.5	23.5	45.5
classifier	79.3	66.2	65.0
posclass	81.4	71.0	66.5
SEQ2SEQ	87.8	75.2	73.7
SEQ2TREE	89.7	**78.4**	74.2
GRU-DRNN	89.9	77.6	74.1
LSTM-DRNN	**90.1**	78.2	**77.4**

Alvarez-Melis, D., Jaakkola, T. S. (2017). Tree-structured decoding with doubly-recurrent neural networks.

Different approaches

Heuristics from other papers

- Introduce special terminal tokens
- 4 independent LSTMs, which act in alternation instead of simultaneously
- Build trees using bottom-up approach
- Concatenating parent and sibling hidden states

Loss function

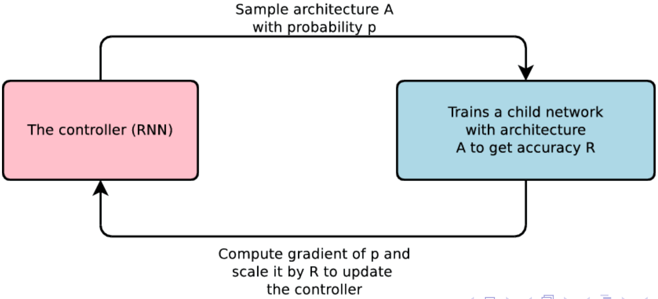
Explicit tree generation + cross-entropy loss

Plan

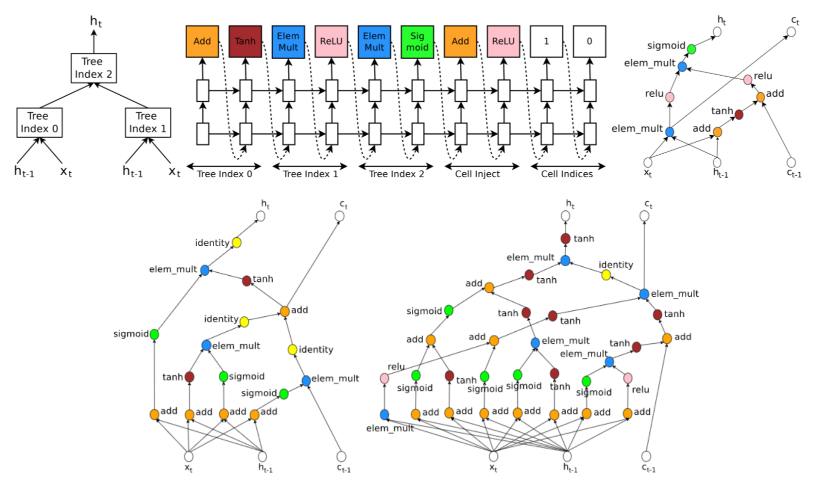
Bonus

Google research

Since May Google Brain team is working on AutoML – an automation of the design of neural networks. They claim that auto-generated neural networks already exceeded state-of-the-art human design for some ML tasks.

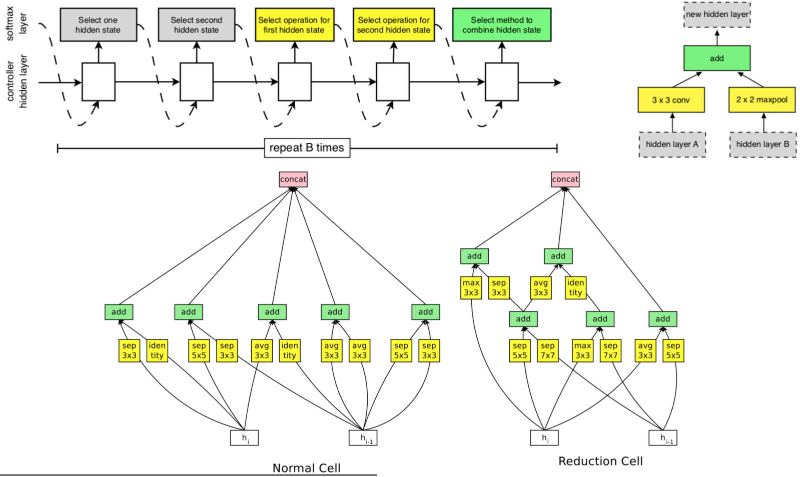


NLP



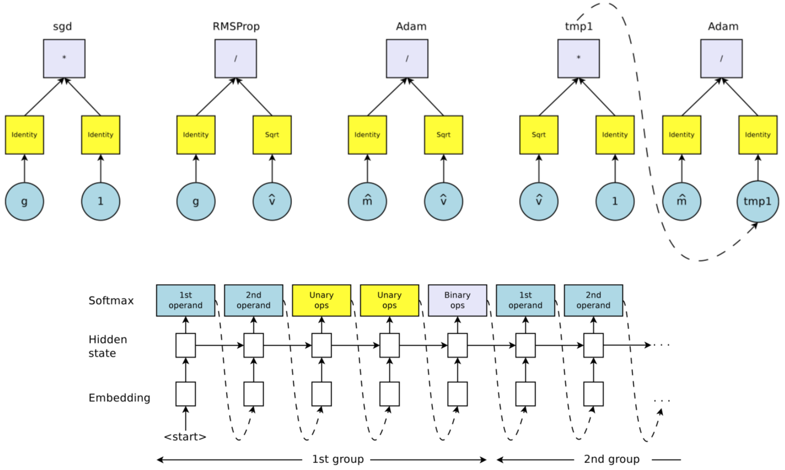
Zoph, B., and Quoc V. Le. Neural architecture search with reinforcement learning. *arXiv preprint* arXiv:1611.01578 (2016).

Image recognition



Zoph, Barret, et al. Learning Transferable Architectures for Scalable Image Recognition. *arXiv preprint* arXiv:1707.07012 (2017).

Optimization methods



Bello, Irwan, et al. Neural optimizer search with reinforcement learning. *arXiv preprint* arXiv:1709.07417 (2017).

Reference I

Alvarez-Melis, D. and Jaakkola, T. S. (2017).
Tree-structured decoding with doubly-recurrent neural networks.

Bello, I., Zoph, B., Vasudevan, V., and Le, Q. V. (2017).
Neural optimizer search with reinforcement learning.
arXiv preprint arXiv:1709.07417.

Dong, L. and Lapata, M. (2016).
Language to logical form with neural attention.
arXiv preprint arXiv:1601.01280.

Reference II

Kulunchakov, A. and Strijov, V. (2017).
Generation of simple structured information retrieval functions by genetic algorithm without stagnation.
Expert Systems with Applications, 85:221–230.

Tai, K. S., Socher, R., and Manning, C. D. (2015).
Improved semantic representations from tree-structured long short-term memory networks.
arXiv preprint arXiv:1503.00075.

Zhang, X., Lu, L., and Lapata, M. (2015).
Top-down tree long short-term memory networks.
arXiv preprint arXiv:1511.00060.

Reference III

Zoph, B. and Le, Q. V. (2016).
Neural architecture search with reinforcement learning.
arXiv preprint arXiv:1611.01578.

Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2017).
Learning transferable architectures for scalable image recognition.
arXiv preprint arXiv:1707.07012.