1 Введение

Тезисы:

- Выбором структуры модели занимается meta-learning
- Мы знаем основные методы порождения моделей, требуется адекватное представление структуры
- Ссылка на обзоры. В целом, мы не касаемся методов порождения обобщенное-линейных моделей
- Текст из Степашко:

  Если речь идет об анализе методов автоматического порождения аппроксимирующих моделей, то можно описать точный круг решаемых задач и ввести общепризнанные базовые математические объекты, с помощью которых попытаться систематизировать и формализовать то множество способов порождения моделей, которые мы имеем. Педро Домингос мог бы это сделать, например (в Master algorithm), но ограничился популярными высказываниями.

2 Метаоптимизация

2.1 Теоретические основания метаобучения

В работе [1] рассматривается задача построения генеративных моделей, предлагается критерий для послойного обучения генеративных моделей:

$$\delta E = \sum_i g_i \delta u_i + \frac{1}{2} \sum_i h_{ii} \delta u_i^2 + \frac{1}{2} \sum_{i \neq j} h_{ij} \delta u_i \delta u_j + O(||\delta^-||^3) \qquad (1)$$

В работе [2] рассматриваются подходы к сэмплированию моделей глубокого обучения. Предлагается формализация пространства поиска и формальное описание эл ементов этого пространства:

```
(Concat
    (Conv2D [32, 64] [3, 5] [1])
    (MaybeSwap BatchNormalization ReLU)
    (Optional (Dropout [0.5, 0.9]))
    (Affine [10]))
```

Figure 1. A simple search space with 24 different models. See Figure 2 for a path through the search space.

2.2 Метаоптимизация: learning to learn

В работе [3] предлагается подход к адаптивному изменению структуры сети, основанный на обучении с подкреплением. Предлагается параметризация модели нейросети, включающая в себя модифицирующие и анализирующие выходы, позволяющие модифицировать параметры модели:

$$net_{y_k}(1) = 0, \quad \forall t \geq 1: \quad x_k(t) \leftarrow environment,$$

$$y_k(t) = f_{y_k}(net_{y_k}(t)),$$

$$\forall t > 1: \quad net_{y_k}(t) = \sum_l w_{y_k l}(t-1) l(t-1), \qquad (7)$$

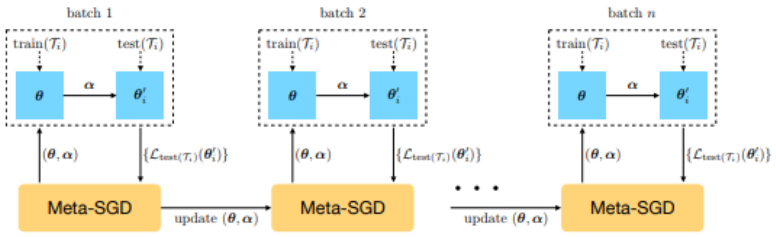
$$\forall t \geq 1: \quad w_{ij}(t+1) = w_{ij}(t) + \triangle(t)\, g[\, \|adr(w_{ij}) - mod(t)\|^2\,] \qquad (8)$$

$$val(1) = 0, \quad \forall t \geq 1: \quad val(t+1) =$$

$$= \sum_{i,j} g[\|ana(t) - adr(w_{ij})\|^2] w_{ij}(t). \qquad (9)$$

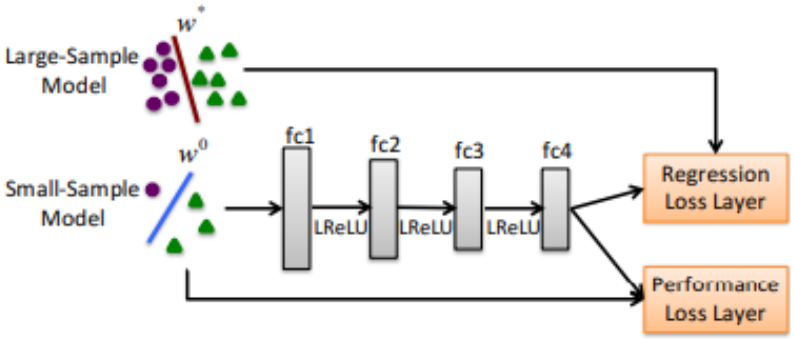
Предлагается продолжение подхода, позволяющая рекуррентно продолжать анализ модели и порождать мета-метан-...-анализ.

В работе [4] рассматривается оптимизация мета-параметров (шага градиентного спуска и начального распределения параметров) с использованием обучения с подкреплением. На каждой итерации сэмплируется подвыборка, по которой проводится оптимизация данных метапараметров:

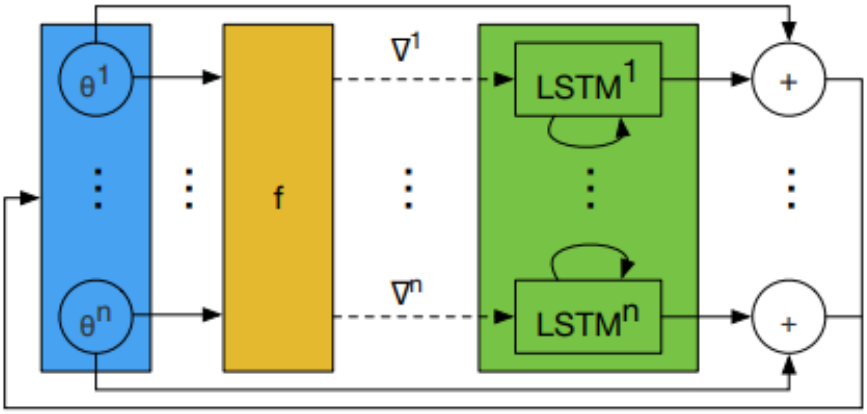


В работе [5] рассматривается задача восстановления параметров модели по параметрам слабо обученной модели:

$$L(\Theta) = \sum_{j=1}^{J} \left\{ \frac{1}{2} \left\| \mathbf{w}_j^* - T\left(\mathbf{w}_j^0, \Theta\right) \right\|_2^2 + \lambda \sum_{i=1}^{M+N} \left[1 - y_i^j \left(T\left(\mathbf{w}_j^0, \Theta\right)^T \mathbf{x}_i^j \right) \right]_+ \right\}. \quad (1)$$



В работе [6] рассматривается оптимизация метапараметров оптимизации с помощью LSTM, которая выступает альтернативе аналитических алгоритмов, таких как Adam или AdaGrad. LSTM имеет (сравнительно) небольшое количество параметров, т.к. для каждого метапараметра используется своя копия модели LSTM с одинаковыми параметрами для каждой копии:



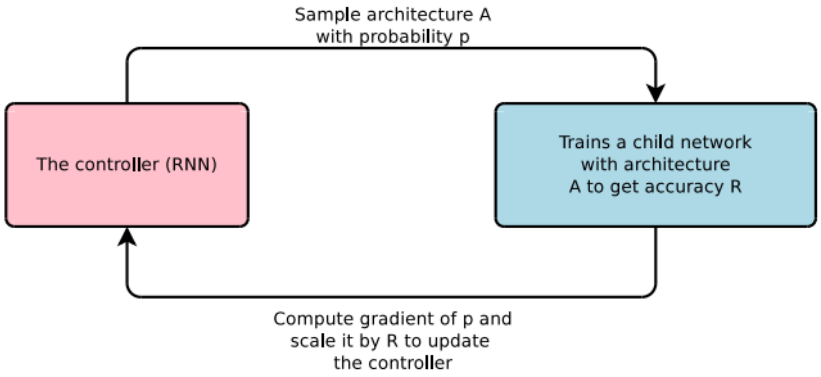
2.3 Перебор структур

В работе [7] рассматривается задача порождения сверточных нейронных сетей. Предлагается проводить поиск оптимальной структуры сети по восходящему по сложности порядку: начиная от сетей с одни блоком и наращивая блоки. В силу высокой вычислительной сложности данного подхода, вместо построения модели,предлагается обучить рекуррентную нейросеть,которая предсказывает качество модели по заданным блокам.

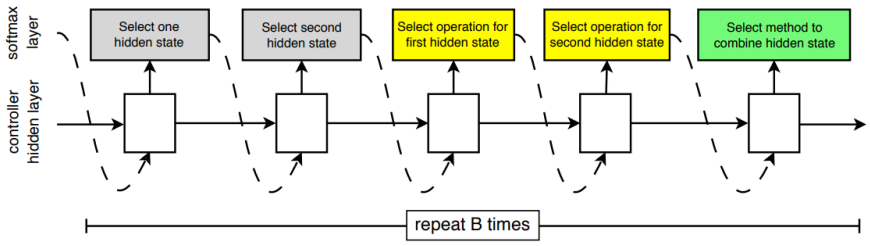
В работе [8] рассматривается задача выбора архитектуры с помощью большого количества параллельных запусков обучения моделей, предлагаются критерии ранней остановки оптимизации обучения моделей.

2.4 Обучение с подкреплением

В работе [9] представлена схема выбора архитектуры сверточной нейросети с использованием обучения с подкреплением. В качестве актора (контроллера) выступает рекуррентная нейронная сеть.

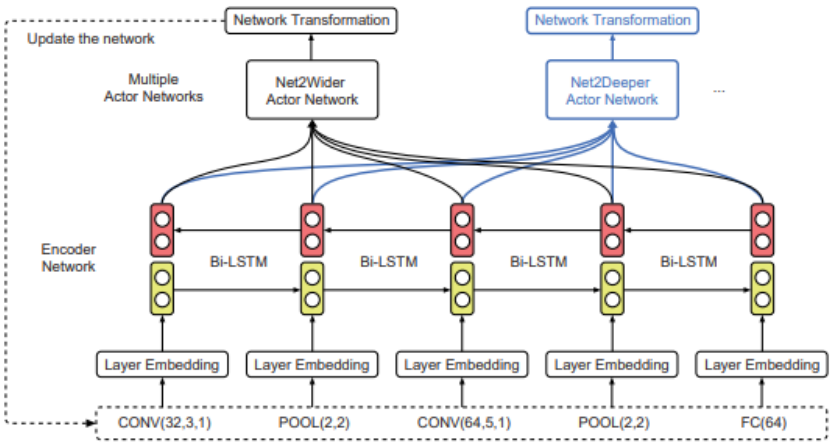


В работе [10] предлагается построение регрессионной модели для оценки финального качества модели и ранней остановки оптимизации моделей. Данный подход позволил существенно ускорить поиск моделей, представленный в работе [9]. В работе [11] рассматривается задача переноса архитектуры нейросети, обученной на более простой выборки, на более сложную. Также предлагается параметризация пространства поиска, более делатьное, чем в [9]:



В отличие от предыдущих работ, в работе [12] предлагается подход к инкрементальному обучению нейросети, основанном на модификации модели, полученной на предыдущем шаге. Рассматривается две операции над нейросетью:

- Расширение сети
- Углубление сети



3 Адаптивное изменение структуры

В данном разделе собраны методы изменения структуры существующей модели.

Алгоритмы наращивания и прореживания параметров модели В работе [13] предлагается удалять неинформативные параметры модели, где в качестве показателя информативности выступает следующий фнуционал:

$$\delta E = \sum_i g_i \delta u_i + \frac{1}{2} \sum_i h_{ii} \delta u_i^2 + \frac{1}{2} \sum_{i \neq j} h_{ij} \delta u_i \delta u_j + O(||\vec{R}||^3) \qquad (1)$$

В работе [14] было предложено развитие данного метода. В данной работе, в отличие от [13] не вводится предположений о дигональности Гессиана функции ошибок, поэтому удаление неинформативных параметров модели производится точнее.

В работе [15] был предложен метод, основанный на получении вариационной нижней оценки правдоподобия модели. В качестве критерия информативности параметра выступало отношение вероятности нахождения параметра в пределах априорного распределения к вероятности равенста параметра нулю:

$$\exp\left(-\frac{\mu_i^2}{2\sigma_i^2}\right) > \gamma \implies \left|\frac{\mu_i}{\sigma_i}\right| < \lambda$$

Идея данного метода была развита в [16], где также используются вариационные методы. В отличие от предыдущей работы, в данной работе рассматривается ряд априорных распределений параметров, позволяющих прореживать модели более эффективно:

- Нормальное распределение с лог-равномерным распределением дисперсии, независимой для каждого нейрона:

$$p(\mathbf{W}, \mathbf{z}) \propto \prod_i^A \frac{1}{|z_i|} \prod_{ij}^{A,B} \mathcal{N}(w_{ij}|0, z_i^2),$$

- Произвденеие двух Распределений Half-Cauchy (полу-Коши?): одно ответственно за отдельный параметр, другое — за общее распределение параметров:

$$s \sim \mathcal{C}^+(0, \tau_0); \qquad \tilde{z}_i \sim \mathcal{C}^+(0, 1); \qquad \tilde{w}_{ij} \sim \mathcal{N}(0, 1); \qquad w_{ij} = \tilde{w}_{ij}\tilde{z}_i s,$$

Смежной темой к прореижванию моделей выступает компрессия нейросетей. Основным отличием задачи прореживания и компрессии выступает эксплуатационное требование: если прореживание используется для получения оптимальной и наиболее устойчивой модели, то компрессия часто производится для сохранения памяти и основных эксплуатационных характеристик исходной модели (?). В работе [17] предлагается итеритавное использование регуляризации типа DropOut [18] для прореживания модели. В работах [19, 20] используются методы снижения вычислительной точности представления парамеров модели на основе кластеризации весов. В работе [20] предлагается метод компрессии, основанный на кластеризации значений параметров модели и представлении их в сжатом виде на основе кодов Хаффмана.

В работах [21, 22] предлагается наращивание моделей, основанное на бустинге. В работе рассматривается задача построения нейросетевых моделей

специального типа:

$$f_{t+1} = \sigma(f_t) + f_t,$$

приводится параметризация модели, позволяющая рассматривать декомпозировать модель на слабые классификаторы. В работе [22] на каждом шаге построения выбирается одно из двух расширений модели, каждое из которых рассматрвается как слабый классификатор: 1. Сделать модель шире 2. Сделать модель глубже

Построение модели заканчивается при условии снижении радемахереовской сложности:

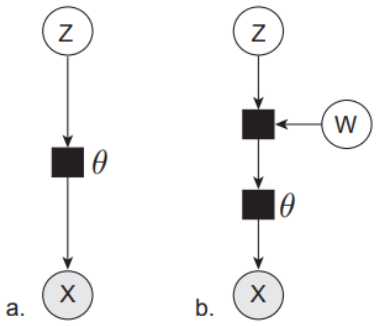
$$\widehat{\mathfrak{R}}_S(\mathcal{G}) = \frac{1}{m}\,\mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{G}}\sum_{i=1}^{m}\sigma_i h(x_i)\right],$$

4 Байесовские методы порождения и выбора моделей

4.1 Автоматическое определение релевантности параметров

В работе [23] рассматривается задача оптимизации гиперпараметров. Авторы предлагают оптимизировать константы l_2-регуляризации отдельно для каждого параметра модели, проводится параллель с методами автоматического определения релевантности параметров (ARD) [24].

В работе [25] рассматривается метод ARD для снижения размерности скрытого пространства вариационных порождающих моделей: скрытая переменная параметризуется как произведение некоторой случайной величины $\mathbf{z}$ на вектор, отвечающий за релевантность каждой компоненты скрытой переменной:



4.2 Суррогаты

В работе [26] предлагается моделировать качество модели гауссовым процессом, параметрами которого выступают гиперпараметры исходной модели. Модель, аппроксимирующая качество исходной модели, называется суррогатом.

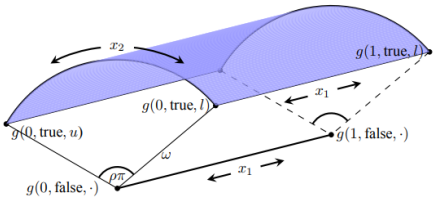
Одна из основных проблем использования гауссового процесса как суррогатной модели — кубическая сложность оптимизации. В работе [27] предлагается использовать случайные подпространства гиперпараметров для ускоренной оптимизации. В работе [28] предлагается кобминация из множества гауссовых моделей и линейной модели, позволяющая модели нелинейные зависимости гиперпараметров, а также существенно сократить сложность оптимизации.

В работе [29] предлагается рассматривать RBF-модель для аппроксимации качества исходной модели, что позволяет ускорить процесс оптимизации суррогатной модели. В [30] рассматривается глубокая нейронная сеть в качестве суррогатной функции. Вместо интеграла правдоподобия, который оценивается в случае использования гауссового процесса в качестве суррогата, используется максимум апостериорной веротяности.

Важным параметром гауссовых процессов является функиия ядра гауссового процесса, полностью определяющая процесс в случае нулевого среднего. В работе [31] предлагается функция ядра, определенная на графах:

$$k(x, y) = r(d(x, y)),$$

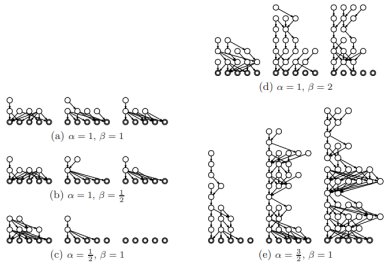
где d — геодезическое расстояние между вершинами графа, r — некоторая вещественная функция (наверно положительно определенная, но это не указано явно в статье). В работе [32] рассматривается задач выбора структуры нейросети, предлагается ядро специального вида, позволюящее учитывать только те гиперпапараметры, которые есть в обеих сравниваемых моделях: к примеру, для двуслойной и трехслойной нейросети будут уччитываться гиперпараметры, отвечающие только за первые два слоя.



4.3 Адаптивное изменение структуры

В работе [33] рассматривается порождение unsupervised-моделей с использованием расширения процесса Индийского Буфета:

$$p(K^{(m+1)} = k \mid K^{(m)}, \alpha, \beta) = \frac{1}{k!} \exp\left\{-\lambda(K^{(m)}; \alpha, \beta)\right\} \lambda(K^{(m)}; \alpha, \beta)^k,$$



В работе [34] предлагается упрощенная модель Индийского Буфета:

$$-\log p(x, W, z) \sim \sum_{i-1}^{N} \|x_i - W z_i\|_2^2 + \lambda^2 K$$

Работы по процессу IBP для порождения unsupervised-моделей (иерархических):

В работе [35] предлагается параметризация структуры модели с использованием Бернуллиевских величин: каждая величина отвечает за включение или выключение слоя сети.

4.4 Порождающие модели

https://arxiv.org/pdf/1406.5298.pdf

https://arxiv.org/pdf/1603.06277.pdf TODO, судя по всему - обобщение вариационного кодировщика на произвольные граф модели

https://arxiv.org/pdf/1611.06585.pdf Предлагается стратегия добавления компонент при вариационном выводе. Проводится аналогия с бустингом.

4.5 Состязательные модели

5 Способы прогнозирования графовых структур

В разделе собраны ключевые работы по порождению графовых моделей.

6 Эвристические и прикладные методы

Прочие работы.

Список литературы

[1] *Arnold Ludovic, Ollivier Yann.* Layer-wise learning of deep generative models // *arXiv preprint arXiv:1212.1524.* — 2012.

[2] *Negrinho Renato, Gordon Geoff.* Deeparchitect: Automatically designing and training deep architectures // *arXiv preprint arXiv:1704.08792.* — 2017.

[3] *Schmidhuber Jürgen.* A neural network that embeds its own meta-levels // Neural Networks, 1993., IEEE International Conference on / IEEE. — 1993. — Pp. 407–412.

[4] Meta-SGD: Learning to Learn Quickly for Few Shot Learning / Zhenguo Li, Fengwei Zhou, Fei Chen, Hang Li // *arXiv preprint arXiv:1707.09835.* — 2017.

[5] *Wang Yu-Xiong, Hebert Martial.* Learning to learn: Model regression networks for easy small sample learning // European Conference on Computer Vision / Springer. — 2016. — Pp. 616–634.

[6] Learning to learn by gradient descent by gradient descent / Marcin Andrychowicz, Misha Denil, Sergio Gomez et al. // Advances in Neural Information Processing Systems. — 2016. — Pp. 3981–3989.

[7] Progressive neural architecture search / Chenxi Liu, Barret Zoph, Jonathon Shlens et al. // *arXiv preprint arXiv:1712.00559.* — 2017.

[8] Toward Optimal Run Racing: Application to Deep Learning Calibration / Olivier Bousquet, Sylvain Gelly, Karol Kurach et al. // *arXiv preprint arXiv:1706.03199.* — 2017.

[9] *Zoph Barret, Le Quoc V.* Neural architecture search with reinforcement learning // *arXiv preprint arXiv:1611.01578.* — 2016.

[10] Accelerating neural architecture search using performance prediction / Bowen Baker, Otkrist Gupta, Ramesh Raskar, Nikhil Naik // *CoRR, abs/1705.10823*. — 2017.

[11] Learning transferable architectures for scalable image recognition / Barret Zoph, Vijay Vasudevan, Jonathon Shlens, Quoc V Le // *arXiv preprint arXiv:1707.07012*. — 2017.

[12] Efficient Architecture Search by Network Transformation / Han Cai, Tianyao Chen, Weinan Zhang et al. — 2018.

[13] *Cun Yann Le, Denker John S., Solla Sara A.* Optimal Brain Damage // Advances in Neural Information Processing Systems. — Morgan Kaufmann, 1990. — Pp. 598–605.

[14] *Hassibi Babak, Stork David G, Wolff Gregory J.* Optimal brain surgeon and general network pruning // Neural Networks, 1993., IEEE International Conference on / IEEE. — 1993. — Pp. 293–299.

[15] *Graves Alex.* Practical Variational Inference for Neural Networks // Advances in Neural Information Processing Systems 24 / Ed. by J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett et al. — Curran Associates, Inc., 2011. — Pp. 2348–2356. http://papers.nips.cc/paper/4329-practical-variational-inference-for-neural-networks.pdf.

[16] *Louizos Christos, Ullrich Karen, Welling Max.* Bayesian compression for deep learning // Advances in Neural Information Processing Systems. — 2017. — Pp. 3290–3300.

[17] Learning both Weights and Connections for Efficient Neural Network / Song Han, Jeff Pool, John Tran, William Dally // Advances in Neural Information Processing Systems 28 / Ed. by C. Cortes, N. D. Lawrence, D. D. Lee et al. — Curran Associates, Inc., 2015. — Pp. 1135–1143. http://papers.nips.cc/paper/5784-learning-both-weights-and-connections-for-efficient-neural-network.pdf.

[18] Dropout: A simple way to prevent neural networks from overfitting / Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky et al. // *The Journal of Machine Learning Research*. — 2014. — Vol. 15, no. 1. — Pp. 1929–1958.

[19] Incremental network quantization: Towards lossless cnns with low-precision weights / Aojun Zhou, Anbang Yao, Yiwen Guo et al. // *arXiv preprint arXiv:1702.03044*. — 2017.

[20] *Han Song, Mao Huizi, Dally William J.* Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding // *arXiv preprint arXiv:1510.00149*. — 2015.

[21] Learning deep resnet blocks sequentially using boosting theory / Furong Huang, Jordan Ash, John Langford, Robert Schapire // *arXiv preprint arXiv:1706.04964*. — 2017.

[22] AdaNet: Adaptive Structural Learning of Artificial Neural Networks / Corinna Cortes, Xavi Gonzalvo, Vitaly Kuznetsov et al. // *CoRR*. — 2016. — Vol. abs/1607.01097. http://arxiv.org/abs/1607.01097.

[23] *Maclaurin Dougal, Duvenaud David, Adams Ryan.* Gradient-based Hyperparameter Optimization through Reversible Learning // Proceedings of the 32nd International Conference on Machine Learning (ICML-15) / Ed. by David Blei, Francis Bach. — JMLR Workshop and Conference Proceedings, 2015. — Pp. 2113–2122. http://jmlr.org/proceedings/papers/v37/maclaurin15.pdf.

[24] *MacKay David J. C.* Information Theory, Inference & Learning Algorithms. — New York, NY, USA: Cambridge University Press, 2002.

[25] *Karaletsos Theofanis, Rätsch Gunnar.* Automatic Relevance Determination For Deep Generative Models // *arXiv preprint arXiv:1505.07765*. — 2015.

[26] *Snoek Jasper, Larochelle Hugo, Adams Ryan P.* Practical bayesian optimization of machine learning algorithms // Advances in neural information processing systems. — 2012. — Pp. 2951–2959.

[27] Bayesian Optimization in High Dimensions via Random Embeddings. / Ziyu Wang, Masrour Zoghi, Frank Hutter et al. // IJCAI. — 2013. — Pp. 1778–1784.

[28] Bayesian Optimization with Tree-structured Dependencies / Rodolphe Jenatton, Cedric Archambeau, Javier González, Matthias Seeger // International Conference on Machine Learning. — 2017. — Pp. 1655–1664.

[29] Hyperparameter optimization of deep neural networks using non-probabilistic RBF surrogate model / Ilija Ilievski, Taimoor Akhtar, Jiashi Feng, Christine Annette Shoemaker // *arXiv preprint arXiv:1607.08316*. — 2016.

[30] Scalable Bayesian Optimization Using Deep Neural Networks / Jasper Snoek, Oren Rippel, Kevin Swersky et al. // Proceedings of the 32nd International Conference on Machine Learning / Ed. by Francis Bach, David Blei. — Vol. 37 of *Proceedings of Machine Learning Research*. — Lille, France: PMLR, 2015. — 07–09 Jul. — Pp. 2171–2180. http://proceedings.mlr.press/v37/snoek15.html.

[31] Structure Optimization for Deep Multimodal Fusion Networks using Graph-Induced Kernels / Dhanesh Ramachandram, Michal Lisicki, Timothy J Shields et al. // *arXiv preprint arXiv:1707.00750*. — 2017.

[32] Raiders of the lost architecture: Kernels for Bayesian optimization in conditional parameter spaces / Kevin Swersky, David Duvenaud, Jasper Snoek et al. // *arXiv preprint arXiv:1409.4011*. — 2014.

[33] *Adams Ryan, Wallach Hanna, Ghahramani Zoubin*. Learning the structure of deep sparse graphical models // Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. — 2010. — Pp. 1–8.

[34] *Feng Jiashi, Darrell Trevor*. Learning the structure of deep convolutional networks // Proceedings of the IEEE international conference on computer vision. — 2015. — Pp. 2749–2757.

[35] *Shirakawa Shinichi, Iwata Yasushi, Akimoto Youhei*. Dynamic Optimization of Neural Network Structures Using Probabilistic Modeling // *arXiv preprint arXiv:1801.07650*. — 2018.