Выбор структуры модели

Московский Физико-Технический Институт

2022

Выбор модели: связанный байесовский вывод

Первый уровень: выбираем оптимальные параметры:

$$\mathbf{w} = \arg\max \frac{p(\mathfrak{D}|\mathbf{w})p(\mathbf{w}|\mathbf{h})}{p(\mathfrak{D}|\mathbf{h})},$$

Второй уровень: выбираем модель, доставляющую максимум обоснованности модели.
Обоснованность модели (“Evidence”):

$$p(\mathfrak{D}|\mathbf{h}) = \int_{\mathbf{w}} p(\mathfrak{D}|\mathbf{w})p(\mathbf{w}|\mathbf{h})d\mathbf{w}.$$

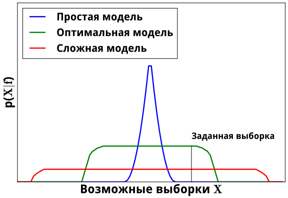
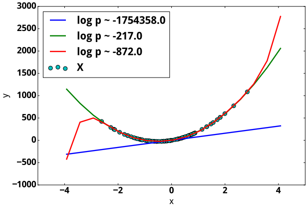


Схема выбора модели

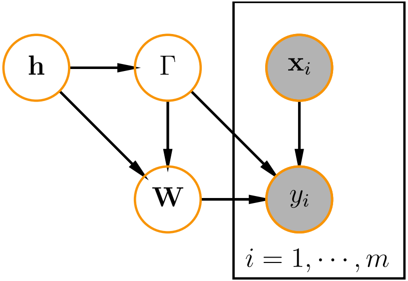


Пример: полиномы

Априорное распределение параметров

Определение

Априорным распределением параметров $\mathbf{w}$ и структуры $\mathbf{\Gamma}$ модели $\mathbf{f}$ назовем вероятностное распределение $p(\mathbf{W}, \mathbf{\Gamma}|\mathbf{h}) : \mathbb{W} \times \mathbb{\Gamma} \times \mathbb{H} \to \mathbb{R}^+$, где $\mathbb{W}$ — множество значений параметров модели, $\mathbb{\Gamma}$ — множество значений структуры модели.



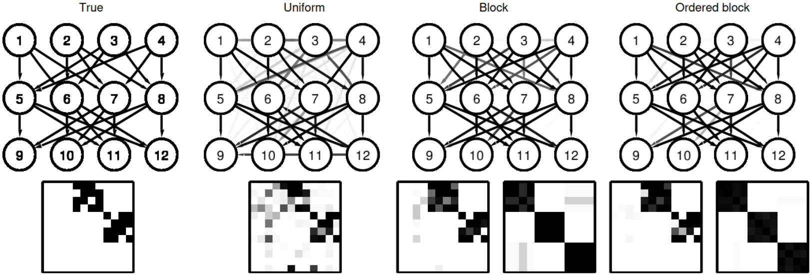
Определение

Гиперпараметрами $\mathbf{h} \in \mathbb{H}$ модели назовем параметры распределения $p(\mathbf{w}, \mathbf{\Gamma}|\mathbf{h})$ (параметры распределения параметров модели $\mathbf{f}$).

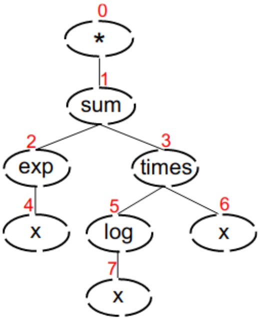
Модель $\mathbf{f}$ задается следующими величинами:

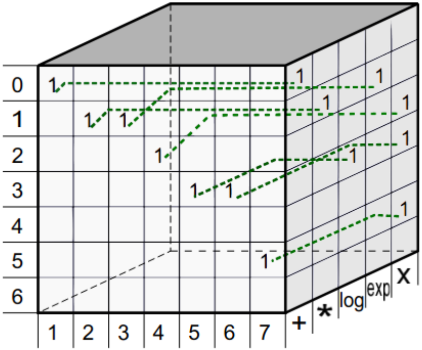
- **Параметры** $\mathbf{w} \in \mathbb{W}$ задают суперпозиции $\mathbf{f}_v$, из которых состоит модель $\mathbf{f}$.
- **Структурные параметры** $\mathbf{\Gamma} = \{\gamma^{j,k}\}_{(j,k)\in E} \in \mathbb{\Gamma}$ задают вклад суперпозиций $\mathbf{f}_v$ в модель $\mathbf{f}$.
- **Гиперпараметры** $\mathbf{h} \in \mathbb{H}$ задают распределение параметров и структурных параметров модели.
- **Метапараметры** $\boldsymbol{\lambda} \in \mathbb{\Lambda}$ задают вид оптимизации модели.

Пример: байесовские сети



Пример: прогнозирование ранжирующей функции





$f = \exp(x) + (\log x)x$

Трехиндексная матрица связей Z_f дерева Γ_f

- вершины дерева пронумерованы;
- первые два индекса — номера вершин в ребре;
- третий индекс — выбранная элементарная функция на конце ребра.

Optimal Brain Damage

Рассматривается задача удаления неинформативных параметров.

Идея метода: Разложим функцию потерь в ряд Тейлора в окрестности максимума $\boldsymbol{\theta}^*$:

$$L(\boldsymbol{\theta}^* + \Delta\boldsymbol{\theta}) - L(\boldsymbol{\theta}^*) = -\frac{1}{2}\boldsymbol{\theta}^\mathsf{T}\mathsf{H}\boldsymbol{\theta} + o(||\Delta\boldsymbol{\theta}||^3),$$

где H — гессиан функции $-L$.

Для простоты вычисления будем полагать гессиан диагональным. Задача удаления параметров сводится к рассмотрению задач условной оптимизации вида:

$$L(\boldsymbol{\theta}^* + \Delta\boldsymbol{\theta}) \to \max$$

при

$$\theta_i^* + \Delta\theta_i = 0.$$

Показатель информативности параметра:

$$\frac{\theta_i^2}{2[\mathsf{H}^{-1}]_{i,i}}.$$

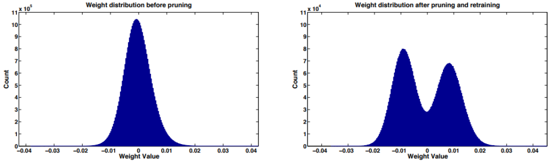
Learning both Weights and Connections for Efficient Neural Networks

Идея подхода:

1. Оптимизируем модель;
2. Удаляем наименьшие по модулю параметры;
3. Запускаем оптимизацию заново.

Почти очевидные факты, которые подтверждаются в статье:

- L_2 лучше для прунинга, чем L_1 в случае, если после прунинга идет оптимизация.
- Оптимизацию лучше производить из предыдущего оптимума, чем из случайной точки.
- После прунинга распределение параметров становится мультимодальным.

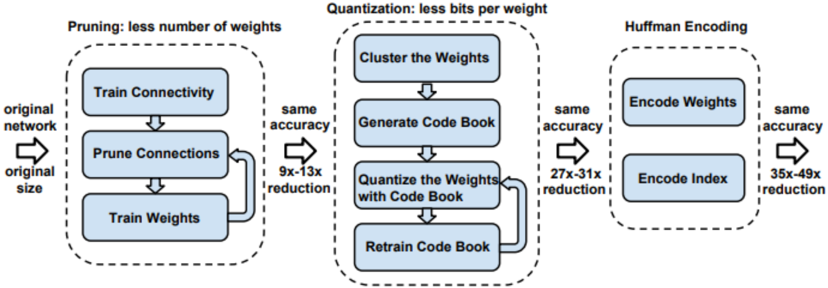


Deep Compression

Идея подхода:

1. Удаляем ненужные параметры модели, аналогично предыдущем подходу.
2. Кластеризуем параметры (K-means на каждом слое).
3. Производим повторную оптимизацию на центроидах.
4. Кодируем индексы параметров с использованием кодов Хаффмана.

Результат: уменьшение размеров модели в 40 раз, ускорение в 3 раза.



Graves, 2011

$$\mathrm{MDL}(\mathrm{f}, \mathfrak{D}) = L(\mathrm{f}) + L(\mathfrak{D}|\mathrm{f}),$$

где f — модель, $\mathfrak{D}$ — выборка, L — длина описания в битах.

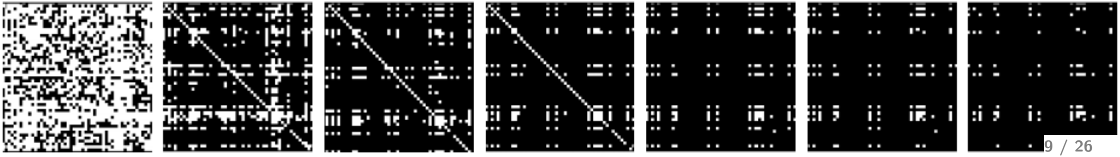
$$\mathrm{MDL}(\mathrm{f}, \mathfrak{D}) \sim L(\mathrm{f}) + L(\mathrm{w}^*|\mathrm{f}) + L(\mathfrak{D}|\mathrm{w}^*, \mathrm{f}),$$

w^* — оптимальные параметры модели.

$$L = \sum_{\mathrm{x},\mathrm{y}} \log p(\mathrm{y}|\mathrm{x}, \hat{\mathrm{w}}) + \frac{1}{2}\left(\mathrm{tr}(\mathrm{A}_q) + \boldsymbol{\mu}_q^{\mathsf{T}} \mathrm{A}^{-1} \boldsymbol{\mu}_q - \ln |\mathrm{A}_q|\right).$$

Прунинг параметра w_i определяется относительной плотностью:

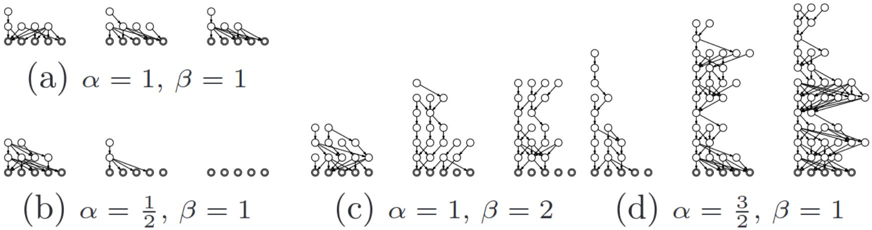
$$\lambda = \frac{q(0)}{q(\boldsymbol{\mu}_{i,q})} = \exp\left(-\frac{\mu_i^2}{2\sigma_i^2}\right).$$



Порождение моделей: пример

Adams et al., 2010:

- Порождаются глубокие сети доверия (Deep belief networks)
- структура модели **Г** — последовательность матриц инцидентности для каждого слоя
- Порождение через Монте-Карло с использованием процесса индийского буффета в качестве априорного с параметрами α, β
- Интерпретация параметров: ширина и разреженность структуры



(a) $\alpha = 1$, $\beta = 1$

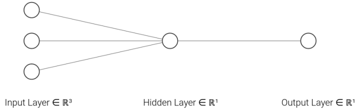
(b) $\alpha = \frac{1}{2}$, $\beta = 1$

(c) $\alpha = 1$, $\beta = 2$

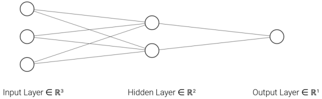
(d) $\alpha = \frac{3}{2}$, $\beta = 1$

Structure selection example

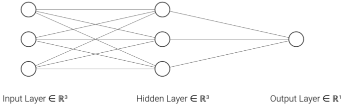
hidden layer dim = 1



hidden layer dim = 2



hidden layer dim = 3



All these models can be represented as $f(x, w) = \boldsymbol{\sigma}\left((w^2)^\mathsf{T} \boldsymbol{\sigma}\left((w^1)^\mathsf{T} x\right)\right)$ with similar shape of $w^1 : \dim(w^1) = 3 \times 3$.

Structure selection: one-layer network

The model f is defined by the **structure** $\boldsymbol{\Gamma} = [\boldsymbol{\gamma}^{0,1}, \boldsymbol{\gamma}^{1,2}]$.

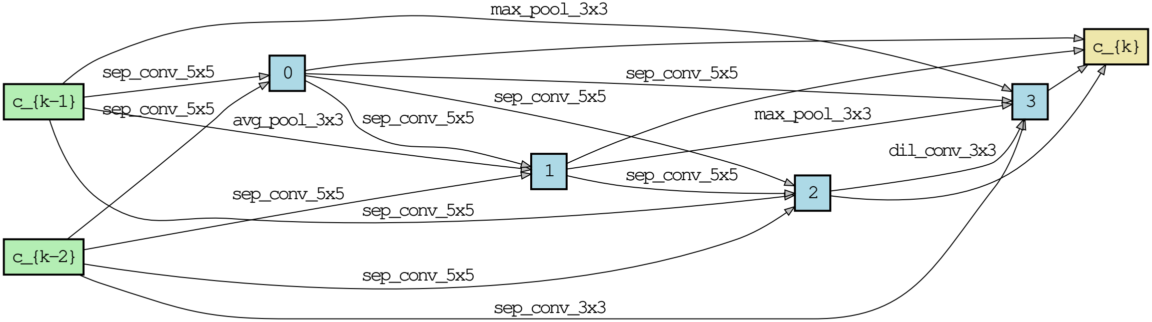
$$\text{Model: } \mathsf{f}(\mathsf{x}) = \textbf{softmax}\left((\mathsf{w}_0^{1,2})^{\mathsf{T}}\mathsf{f}_1(\mathsf{x})\right), \quad \mathsf{f}(\mathsf{x}) : \mathbb{R}^n \to [0,1]^{|\mathbb{Y}|}, \quad \mathsf{x} \in \mathbb{R}^n.$$

$$\mathsf{f}_1(\mathsf{x}) = \gamma_0^{0,1}\mathsf{g}_0^{0,1}(\mathsf{x}) + \gamma_1^{0,1}\mathsf{g}_1^{0,1}(\mathsf{x}),$$

where $\mathsf{w} = [\mathsf{w}_0^{0,1}, \mathsf{w}_1^{0,1}, \mathsf{w}_0^{1,2}]^{\mathsf{T}}$ — parameter matrices, $\{\mathsf{g}_{0,1}^0, \mathsf{g}_{0,1}^1, \mathsf{g}_{1,2}^0\}$ — generalized-linear functions, alternatives of layers of the network.

$$\begin{tikzcd}
\mathsf{f}_0(\mathsf{x}) = \mathsf{x} \arrow[r, bend left, "\gamma_0^{0,1}\mathsf{g}_0^{0,1}(\mathsf{x}) = \gamma_0^{0,1}\boldsymbol{\sigma}\left((\mathsf{w}_0^{0,1})^{\mathsf{T}}\mathsf{x}\right)"] \arrow[r, bend right, "\gamma_1^{0,1}\mathsf{g}_1^{0,1}(\mathsf{x}) = \gamma_1^{0,1}\boldsymbol{\sigma}\left((\mathsf{w}_1^{0,1})^{\mathsf{T}}\mathsf{x}\right)"'] & \mathsf{f}_1(\mathsf{x}) \arrow[r, "\gamma_0^{1,2}\mathsf{g}_0^{1,2}(\mathsf{x}) = \gamma_0^{1,2}\textbf{softmax}\left((\mathsf{w}_0^{1,2})^{\mathsf{T}}\mathsf{x}\right)"] & \mathsf{f}_2(\mathsf{x})
\end{tikzcd}$$

Neural architecture search example



Structure selection: neural architecture search space

The model f is defined by the **structure** $\boldsymbol{\Gamma} = [\boldsymbol{\gamma}^{0,1}, \boldsymbol{\gamma}^{1,2}]$.

$$\text{Model: } \mathrm{f}(\mathrm{x}) = \textbf{softmax}\left((\mathrm{w}_0^{1,2})^{\mathsf{T}}\mathrm{f}_1(\mathrm{x})\right), \quad \mathrm{f}(\mathrm{x}) : \mathbb{R}^n \to [0,1]^{|\mathbb{Y}|}, \quad \mathrm{x} \in \mathbb{R}^n.$$

$$\mathrm{f}_1(\mathrm{x}) = \gamma_0^{0,1}\mathrm{g}_0^{0,1}(\mathrm{x}) + \gamma_1^{0,1}\mathrm{g}_1^{0,1}(\mathrm{x}),$$

where $\mathrm{w} = [\mathrm{w}_0^{0,1}, \mathrm{w}_0^{1,2}]^{\mathsf{T}}$ — parameter matrices, $\mathrm{g}_{0,1}^0$ is a convolution,$\mathrm{g}_{0,1}^1$ is a pooling operation, $\mathrm{g}_{1,2}^0$ is a generalized-linear function.

$$\begin{tikzcd}[column sep=huge]
\mathrm{f}_0(\mathrm{x}) = \mathrm{x} \arrow[r, bend left, "\gamma_0^{0,1}\mathrm{g}_0^{0,1}(\mathrm{x}) = \gamma_0^{0,1}\textbf{Conv}(\mathrm{x}, \mathrm{w}_0^{0,1})"] \arrow[r, bend right, "\gamma_1^{0,1}\mathrm{g}_1^{0,1}(\mathrm{x}) = \gamma_1^{0,1}\textbf{MaxPool}(\mathrm{x})"'] & \mathrm{f}_1(\mathrm{x}) \arrow[r, "\gamma_0^{1,2}\mathrm{g}_0^{1,2}(\mathrm{x}) = \gamma_0^{1,2}\textbf{softmax}\left((\mathrm{w}_0^{1,2})^{\mathsf{T}}\mathrm{x}\right)"] & \mathrm{f}_2(\mathrm{x})
\end{tikzcd}$$

Deep learning model structure as a graph

Define:

1. acyclic graph (V, E);
2. for each edge $(j, k) \in E$: a vector primitive differentiable functions $\mathrm{g}^{j,k} = [\mathrm{g}_0^{j,k}, \dots, \mathrm{g}_{K^{j,k}}^{j,k}]$ with length of $K^{j,k}$;
3. for each vertex $v \in V$: a differentiable aggregation function $\mathbf{agg}_v$.
4. a function $\mathrm{f} = \mathrm{f}_{|V|-1}$:

$$\mathrm{f}_v(\mathrm{w}, \mathrm{x}) = \mathbf{agg}_v \left(\{ \langle \boldsymbol{\gamma}^{j,k}, \mathrm{g}^{j,k} \rangle \circ \mathrm{f}_j(\mathrm{x}) | j \in \mathrm{Adj}(v_k) \} \right), v \in \{1, \dots, |V| - 1\}, \quad \mathrm{f}_0(\mathrm{x}) = \mathrm{x} \tag{1}$$

that is a function from $\mathbb{X}$ into a set of labels $\mathbb{Y}$ for any value of $\boldsymbol{\gamma}^{j,k} \in [0, 1]^{K^{j,k}}$.

Definition

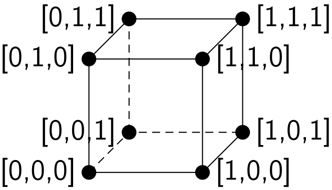
A *parametric set of models* $\mathfrak{F}$ is a graph (V, E) with a set of primitive functions $\{\mathrm{g}^{j,k}, (j, k) \in E\}$ and aggregation functions $\{\mathbf{agg}_v, v \in V\}$.

Statement

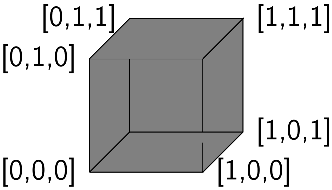
A function $\mathrm{f} \in \mathfrak{F}$ is a model for each $\boldsymbol{\gamma}^{j,k} \in [0, 1]^{K^{j,k}}$.

Structure restrictions

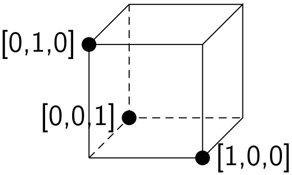
An example of restrictions for structure parameter γ, $|\gamma| = 3$.



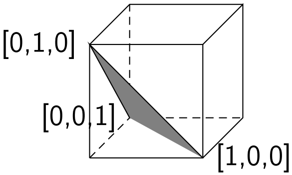
Cube vertices



Cube interior



Simplex vertices

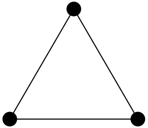


Simplex interior

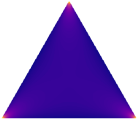
Prior distribution for the model structure

Every point in a simplex defines a model.

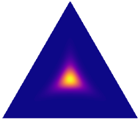
Gumbel-Softmax distribution: $\Gamma \sim \mathrm{GS}(s, \lambda_{\text{temp}})$



$\lambda_{\text{temp}} \to 0$

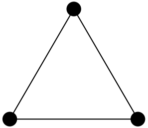


$\lambda_{\text{temp}} = 0.995$

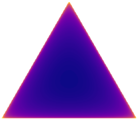


$\lambda_{\text{temp}} = 5.0$

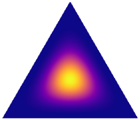
Dirichlet distribution: $\Gamma \sim \mathrm{Dir}(s, \lambda_{\text{temp}})$



$\lambda_{\text{temp}} \to 0$



$\lambda_{\text{temp}} = 0.995$



$\lambda_{\text{temp}} = 5.0$

Neural Architecture Search: постановка задачи

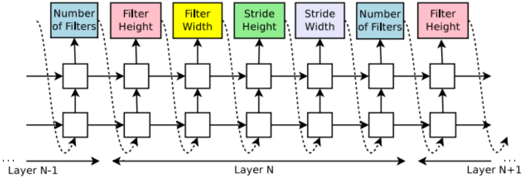
$\boldsymbol{w}$ — параметры модели, оптимизируемые при заданной структуре.
$\boldsymbol{\Gamma}$ — структура модели, задается контроллером, должна доставлять максимум валидации.

$$\boldsymbol{\Gamma}^* = \arg\max Q(\boldsymbol{w}^*, \boldsymbol{\Gamma}),$$

$$\boldsymbol{w}^* = \arg\max L(\boldsymbol{w}, \boldsymbol{\Gamma}).$$

Neural Architecture Search with Reinforcement Learning

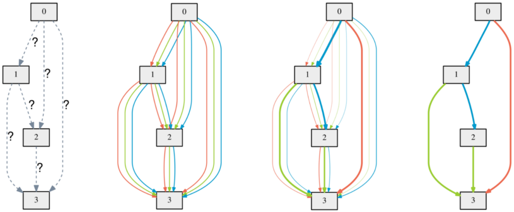
Структура выбирается контроллером. В цикле выбора структуры производится полная оптимизация параметров модели.



DARTS

Модель — мультиграф, где ребра $[g^e]$ соответствуют подмоделям, а вершины $f_v(x)$ — результату действия подмоделей на выборку.
Результат применения подмоделей:

$$f_v = \langle \gamma, \mathbf{softmax}([g^e(x)]) \rangle.$$



DARTS

Задача оптимизации:

$$\mathbf{\Gamma}^* = \arg\max Q(\boldsymbol{w}^*, \mathbf{\Gamma}),$$

$$\boldsymbol{w}^* = \arg\max L(\boldsymbol{w}, \mathbf{\Gamma}).$$

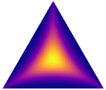
Оптимизация структуры производится жадным градиентным методом:

$$\nabla_{\mathbf{\Gamma}} Q(\boldsymbol{w}', \mathbf{\Gamma}) = \lambda_L \nabla_{\mathbf{\Gamma}, \boldsymbol{w}} L(\boldsymbol{w}, \mathbf{\Gamma}) \nabla_{\boldsymbol{w}} Q(\mathbf{\Gamma}, \boldsymbol{w}').$$

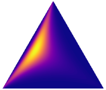
Перебор структур

Процесс перебора можно осуществить путем добавления регуляризации на структуру:

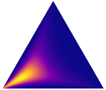
$$\lambda_1 \mathrm{KL}(\boldsymbol{\Gamma}|\boldsymbol{\Gamma}_1) + \lambda_2 \mathrm{KL}(\boldsymbol{\Gamma}|\boldsymbol{\Gamma}_2) + \dots$$



$\boldsymbol{\lambda}_{\text{struct}} = [0; 0; 0].$



$\boldsymbol{\lambda}_{\text{struct}} = [1; 0; 0].$



$\boldsymbol{\lambda}_{\text{struct}} = [1; 1; 0].$

Эксплуатационные критерии качества для выбора структуры

- Количество параметров в элементах структуры
- Количество вершин в структуре
- Количество ребер в структуре
- Сложность вычисления подфункции

FBNet

$$\min_{\boldsymbol{\Gamma}} \min_{\boldsymbol{w}} L \cdot \lambda_1 \log^{\lambda_2} \mathrm{LAT}(\boldsymbol{\Gamma}),$$

где LAT — функция аппаратной задержки операций в структуре, измеренная для **целевого железа**.

FBNet

Model	#Parameters	#FLOPs	Latency on iPhone X	Latency on Samsung S8	Top-1 acc (%)
FBNet-iPhoneX	4.47M	322M	19.84 ms (target)	23.33 ms	73.20
FBNet-S8	4.43M	293M	27.53 ms	22.12 ms (target)	73.27

Table 5. FBNets searched for different devices.

Литература

- Bishop C. M., Nasrabadi N. M. Pattern recognition and machine learning. – New York : springer, 2006. – Т. 4. – №. 4. – С. 738.
- Бахтеев О. Ю. Байесовский выбор субоптимальной структуры модели глубокого обучения, диссертация
- Mansinghka V. et al. Structured priors for structure learning //arXiv preprint arXiv:1206.6852. – 2012.
- Варфоломеева А. А. Методы структурного обучения в задаче обобщения структур прогностических моделей, магистерская диссертация
- LeCun Y., Denker J., Solla S. Optimal brain damage //Advances in neural information processing systems. – 1989. – Т. 2.
- Han S. et al. Learning both weights and connections for efficient neural network //Advances in neural information processing systems. – 2015. – Т. 28.
- Graves A. Practical variational inference for neural networks //Advances in neural information processing systems. – 2011. – Т. 24.
- Han S., Mao H., Dally W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding //arXiv preprint arXiv:1510.00149. – 2015.
- Adams R. P., Wallach H., Ghahramani Z. Learning the structure of deep sparse graphical models //Proceedings of the thirteenth international conference on artificial intelligence and statistics. – JMLR Workshop and Conference Proceedings, 2010. – С. 1-8.
- Jang E., Gu S., Poole B. Categorical reparameterization with gumbel-softmax //arXiv preprint arXiv:1611.01144. – 2016.
- Zoph B., Le Q. V. Neural architecture search with reinforcement learning //arXiv preprint arXiv:1611.01578. – 2016.
- Liu H., Simonyan K., Yang Y. Darts: Differentiable architecture search //arXiv preprint arXiv:1806.09055. – 2018.
- Wu B. et al. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2019. – С. 10734-10742.