Bayesian selection of deep learning model structure

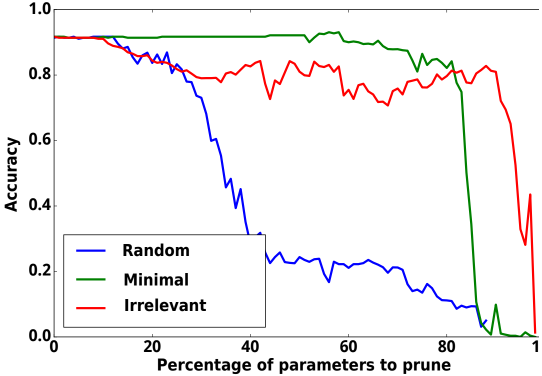
Oleg Bakhteev, Vadim Strijov

Moscow Institute of Physics and Technology

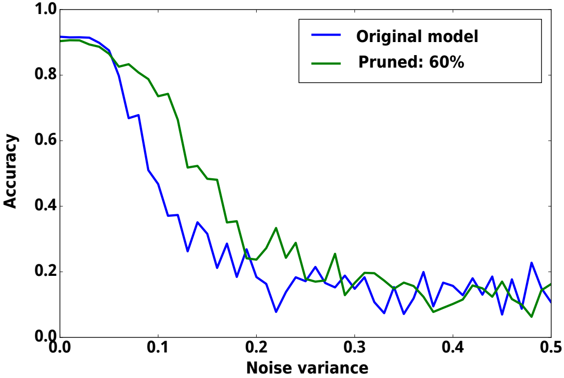
July 2, 2021

Model structure selection challenge

Data likelihood does not change with removing redundant parameters.



Redundancy of model parameters



Model robustness

Deep learning models have implicitly redundant complexity.

Deep learning model

Definition

Model $\mathrm{f}(\mathrm{w}, \mathrm{x})$ is a differentiable function with respect to parameters w from the set of object descriptions into the set of labels:

$$\mathrm{f} : \mathbb{X} \times \mathbb{W} \to \mathbb{Y},$$

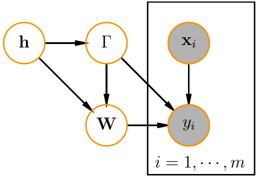
where $\mathbb{W}$ is a space of parameters of model f.

Main challenge of deep learning model selection is in large number of parameters of models. This disallows to use many classical approached for the model and structure selection (AIC, BIC, cross-validation).

A model is defined by its parameters W and structure Γ.
A **structure** defines a set of functional superpositions in the model. It is selected using statistical complexity criteria.
Empirical model complexity estimations:

1. number of parameters;
2. number of superpositions in the model.

Prior distribution



Definition

Prior distribution for parameters $\mathbf{w}$ and structure Γ of model $\mathbf{f}$ is a distribution $p(\mathbf{W}, \Gamma | \mathbf{h}, \boldsymbol{\lambda}) : \mathbb{W} \times \mathbb{\Gamma} \times \mathbb{H} \to \mathbb{R}^+$, where $\mathbb{W}$ is a parameter space, $\mathbb{\Gamma}$ is a structure space, $\boldsymbol{\lambda}$ is a vector of metaparameters.

Definition

Hyperparameters $\mathbf{h} \in \mathbb{H}$ are the parameters of prior distribution $p(\mathbf{w}, \Gamma | \mathbf{h}, \mathbf{f})$ (parameters of the distribution of the parameters and structure of model $\mathbf{f}$).

A model $\mathbf{f}$ is defined by:

- **Parameters** $\mathbf{w} \in \mathbb{W}$ that define superpositions $\mathbf{f}_v$ in the model $\mathbf{f}$.
- **Structure** $\Gamma = \{\gamma^{j,k}\}_{(j,k) \in E} \in \mathbb{\Gamma}$ that define the contribution of all the superpositions $\mathbf{f}_v$ into $\mathbf{f}$.
- **Hyperparameters** $\mathbf{h} \in \mathbb{H}$ that define the prior distribution.
- **Metaparameters** $\boldsymbol{\lambda} \in \mathbb{\Lambda}$ that define the optimization function.

Structure selection: one-layer network

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

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

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

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.

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

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

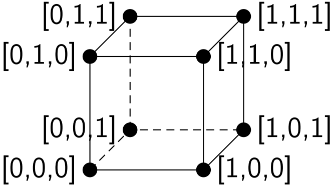
A *parametric set of models* $\mathfrak{F}$ is a graph (V, E) with a set of primitive functions $\{\mathrm{g}^{j,k}, (j, k) \in E\}$ and aggregation functions $\{\mathbf{agg}_v, v \in V\}$.

Statement

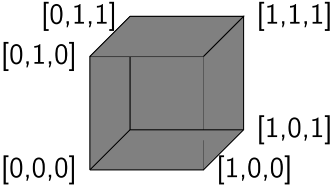
A function $\mathrm{f} \in \mathfrak{F}$ is a model for each $\boldsymbol{\gamma}^{j,k} \in [0,1]^{K^{j,k}}$.

Structure restrictions

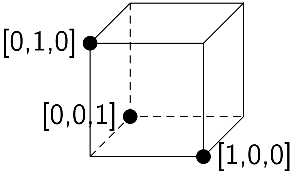
An example of restrictions for structure parameter γ, $|\gamma| = 3$.



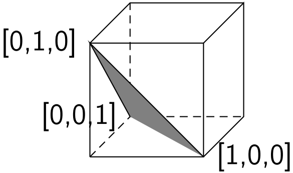
Cube vertices



Cube interior



Simplex vertices

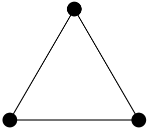


Simplex interior

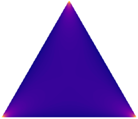
Prior distribution for the model structure

Every point in a simplex defines a model.

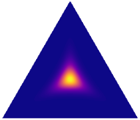
Gumbel-Softmax distribution: $\Gamma \sim \mathrm{GS}(s, \lambda_{\text{temp}})$



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

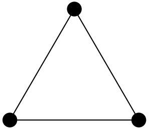


$\lambda_{\text{temp}} = 0.995$

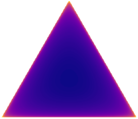


$\lambda_{\text{temp}} = 5.0$

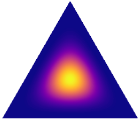
Dirichlet distribution: $\Gamma \sim \mathrm{Dir}(s, \lambda_{\text{temp}})$



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



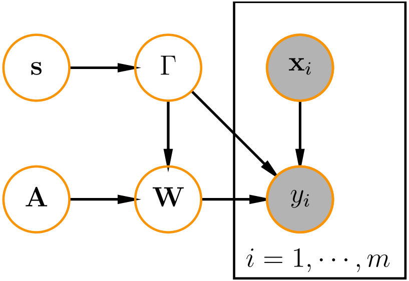
$\lambda_{\text{temp}} = 0.995$



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

Bayesian model selection

- **parameters**
  $\mathrm{w}_r^{j,k} \sim \mathcal{N}(0, (\gamma_r^{j,k})^2 (\mathrm{A}_r^{j,k})^{-1})$, $\mathrm{A}_r^{j,k}$ is a diagonal matrix for the parameters of the primitive function $\mathrm{g}_r^{j,k}$,
- **structure**
  $\Gamma = \{\gamma^{j,k}, (j,k) \in E\}$,
  $\gamma^{j,k} \sim \mathrm{GS}(\mathrm{s}^{j,k}, \lambda_{\mathrm{temp}})$,
- **hyperparameters** $\mathrm{h} = [\mathrm{diag}(\mathrm{A}), \mathrm{s}]$,
- **metaparameters** λ_{temp}.



Evidence as a statistical complexity

Minimum description length for the model f:

$$\mathrm{MDL}(\mathrm{y}, \mathrm{f}) = -\log\, p(\mathrm{h}|\mathrm{f}) - \log\, p(\hat{\mathrm{w}}|\mathrm{h}, \mathrm{f}) - \log\, \left(p(\mathrm{y}|\mathrm{X}, \hat{\mathrm{w}}, \mathrm{f})\delta\mathfrak{D}\right),$$

where $\delta\mathfrak{D}$ is an information transmission precision.

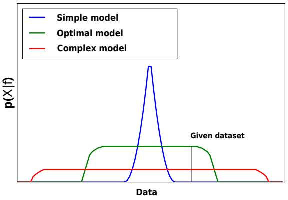
Bayesian approach:

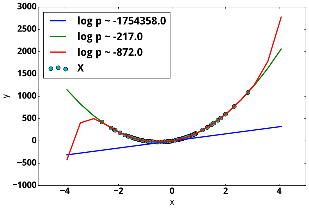
Obtain values of parameters w with respect to **posterior distribution of parameters**:

$$L = \log p(\mathrm{w}|\mathrm{X}, \mathrm{y}, \mathrm{h}, \boldsymbol{\lambda}) \propto \log p(\mathrm{y}|\mathrm{X}, \mathrm{w}, \mathrm{h}, \boldsymbol{\lambda}) + \log p(\mathrm{w}|\mathrm{h}, \boldsymbol{\lambda}).$$

Hyperparameters are optimized using **posterior distribution of hyperparameters**:

$$Q = \log p(\mathrm{f}|\mathrm{X}, \mathrm{y}) \propto \log p(\mathrm{h}|\mathrm{f}) + \log \int\limits_{\mathrm{w}} p(\mathrm{y}|\mathrm{X}, \mathrm{w}, \boldsymbol{\lambda})p(\mathrm{w}|\mathrm{h}, \boldsymbol{\lambda})d\mathrm{w}.$$





Evidence lower bound

The evidence is analytically intractable.

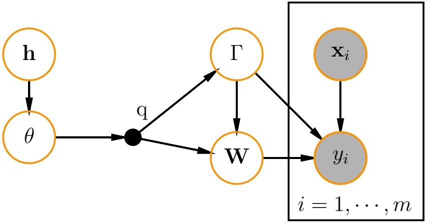
Model evidence:

$$p(\mathrm{y}|\mathrm{X},\mathrm{h},\boldsymbol{\lambda}) = \iint\limits_{\mathrm{w},\Gamma} p(\mathrm{y}|\mathrm{X},\mathrm{w},\Gamma)p(\mathrm{w},\Gamma|\mathrm{h},\boldsymbol{\lambda})d\mathrm{w}d\Gamma.$$

Definition

Variational parameters of the model $\boldsymbol{\theta} \in \mathbb{\Theta}$ are the parameters of the distribution q that approximates posterior distribution $p(\mathrm{w},\Gamma|\mathrm{X},\mathrm{y},\mathrm{h},\boldsymbol{\lambda})$:

$$q \approx \frac{p(\mathrm{y}|\mathrm{X},\mathrm{w},\Gamma)p(\mathrm{w},\Gamma|\mathrm{h},\boldsymbol{\lambda})}{\iint\limits_{\mathrm{w}',\Gamma'} p(\mathrm{y}|\mathrm{X},\mathrm{w}',\Gamma')p(\mathrm{w}',\Gamma'|\mathrm{h},\boldsymbol{\lambda})d\mathrm{w}'d\Gamma'}.$$



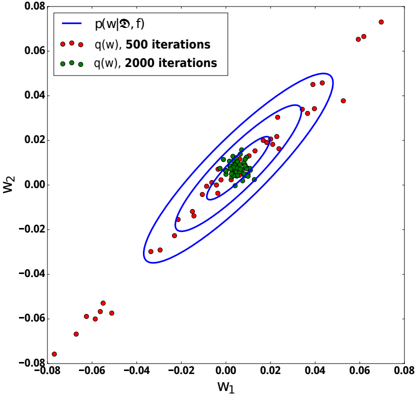
Lower bound of $\log p(\mathrm{y}|\mathrm{X},\mathrm{h},\boldsymbol{\lambda})$:

$$\log p(\mathrm{y}|\mathrm{X},\mathrm{h},\boldsymbol{\lambda}) \geq \mathsf{E}_q \log p(\mathrm{y}|\mathrm{X},\mathrm{w},\Gamma) - \mathsf{D}_{KL}\big(q(\mathrm{w},\Gamma)||p(\mathrm{w},\Gamma|\mathrm{h},\boldsymbol{\lambda})\big).$$

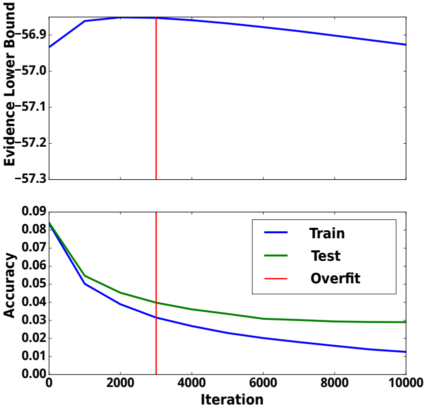
Gradient descent as an evidence lower bound

Empirical distribtuion of the optimized model parameters is a variational distribution.

Gradient descent does not optimize evidence lower bound.



Evidence lower bound decrease is a signal of overfitting.



Model selection problem

Define a variational distribution $q = q_{\mathrm{w}} q_{\Gamma}$ with parameters $\boldsymbol{\theta}$ that approximates posterior distribution $p(\mathrm{w}, \Gamma|\mathrm{X}, \mathrm{y}, \mathrm{h}, \mathrm{f})$.

Definition

Loss function $L(\boldsymbol{\theta}|\mathrm{y}, \mathrm{X}, \mathrm{h}, \boldsymbol{\lambda})$ is a differentiable function interpreted as a performance of the model on the train dataset.

Validation function $Q(\mathrm{h}|\mathrm{y}, \mathrm{X}, \boldsymbol{\theta}, \boldsymbol{\lambda})$ is a differentiable function interpreted as a general performance of the model.

The *model selection problem* f is a level optimization:

$$\mathrm{h}^* = \arg\max_{\mathbf{h} \in \mathbb{H}} Q(\mathrm{h}|\mathrm{y}, \mathrm{X}, \boldsymbol{\theta}^*, \boldsymbol{\lambda}),$$

where $\boldsymbol{\theta}^*$ is a solution for the following optimization:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta} \in \mathbb{U}} L(\boldsymbol{\theta}|\mathrm{y}, \mathrm{X}, \mathrm{h}, \boldsymbol{\lambda}).$$

Proposed optimization problem

Theorem [Bakhtreev, 2019]

The following problem is generalizing:

$$\mathrm{h}^* = \arg\max_{\mathrm{h}} Q =$$

$$= \lambda^{\mathrm{Q}}_{\text{likelihood}} \mathsf{E}_{q(\mathrm{w},\Gamma|\boldsymbol{\theta}^*)} \log\ p(\mathrm{y}|\mathrm{X},\mathrm{w},\Gamma,\mathrm{h},\boldsymbol{\lambda}) -$$

$$-{}^{\text{prior}}_{\mathrm{Q}}\mathrm{D}_{KL}\big(q(\mathrm{w},\Gamma|\boldsymbol{\theta}^*)||p(\mathrm{w},\Gamma|\mathrm{h},\boldsymbol{\lambda})\big) -$$

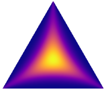
$$-\sum_{p' \in \mathfrak{P}, \lambda \in \boldsymbol{\lambda}^{\text{struct}}_{\mathbf{Q}}} \lambda \mathrm{D}_{KL}\big(p(\Gamma|\mathrm{h},\boldsymbol{\lambda})|p'\big) + \log p(\mathrm{h}|\boldsymbol{\lambda}),$$

where

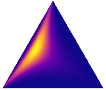
$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} L = \mathsf{E}_q \log\ p(\mathrm{y}|\mathrm{X},\mathrm{w},\Gamma,\mathrm{h},\boldsymbol{\lambda})$$

$$-{}^{\text{prior}}_{\mathrm{L}}\mathrm{D}_{KL}\big(q^*(\mathrm{w},\Gamma)||p(\mathrm{w},\Gamma|\mathrm{h},\boldsymbol{\lambda})\big).$$

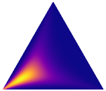
The proposed optimization generalized different optimization problems: maximum likelihood and evidence lower bound optimization, model complexity increase and decrease, exhaustive structure search.



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



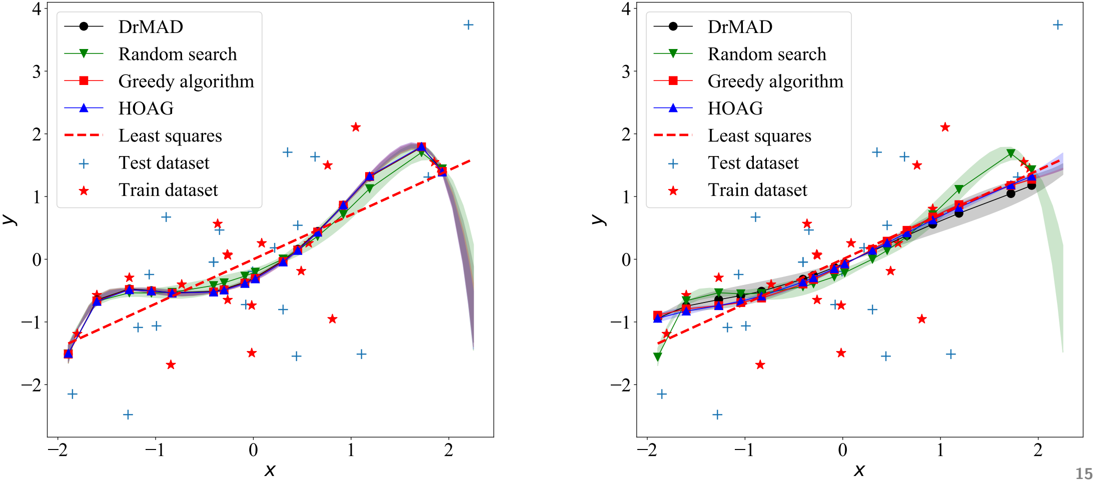
$\boldsymbol{\lambda}^{Q}_{\text{struct}} = [1; 0; 0].$



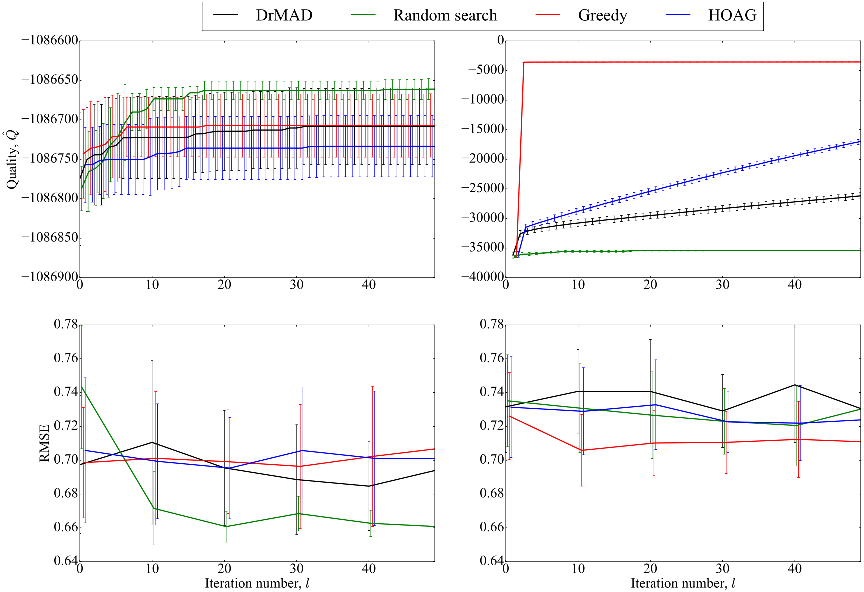
$\boldsymbol{\lambda}^{Q}_{\text{struct}} = [1; 1; 0].$

Hyperparameter optimization: example

Toy example: polynomial regression with potential overfitting.



Experiments: WISDM



Toy example

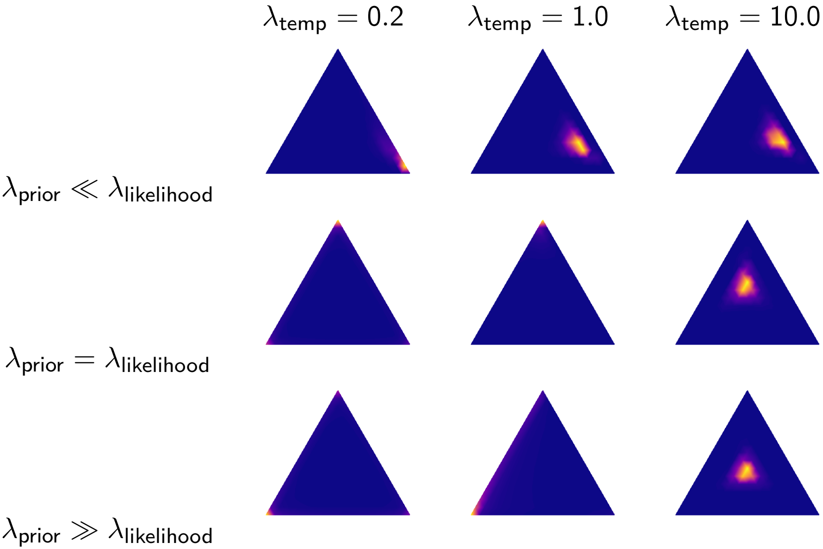
A model f is an ensebmle of 3 models:

1. $\mathrm{g}_0^{0,1} = tanh(wx)$;
2. $\mathrm{g}_1^{0,1} = tanh(\mathrm{w}^\mathsf{T}[x, x^2, \dots, x^{10}])$;
3. $\mathrm{g}_2^{0,1} = w$.

The optimization ran with three regimes:

- $\lambda_{\text{prior}} \ll \lambda_{\text{likelihood}}$;
- $\lambda_{\text{prior}} = \lambda_{\text{likelihood}}$;
- $\lambda_{\text{prior}} \gg \lambda_{\text{likelihood}}$;

Toy dataset: structures



$\lambda_{\text{temp}} = 0.2$

$\lambda_{\text{temp}} = 1.0$

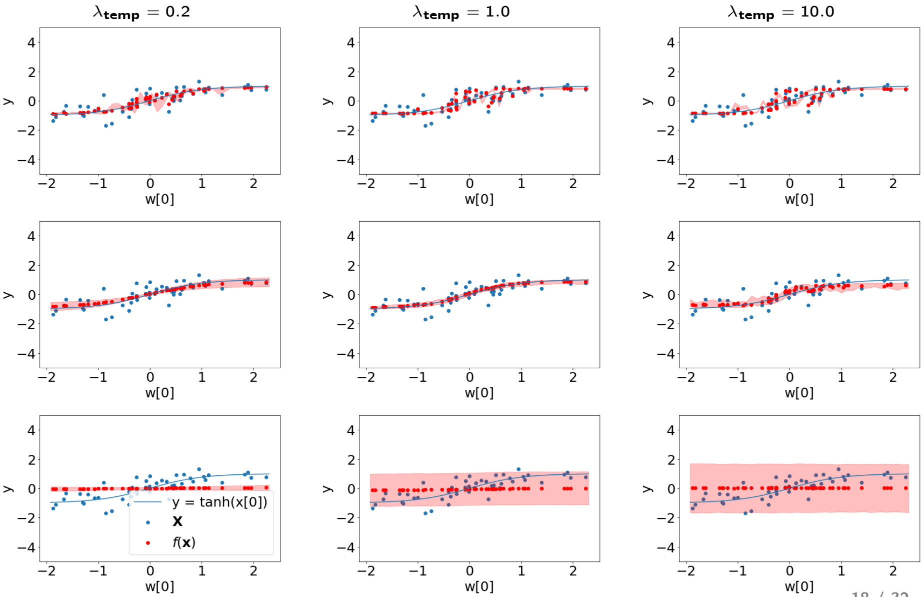
$\lambda_{\text{temp}} = 10.0$

$\lambda_{\text{prior}} \ll \lambda_{\text{likelihood}}$

$\lambda_{\text{prior}} = \lambda_{\text{likelihood}}$

$\lambda_{\text{prior}} \gg \lambda_{\text{likelihood}}$

Toy dataset: prediction performance



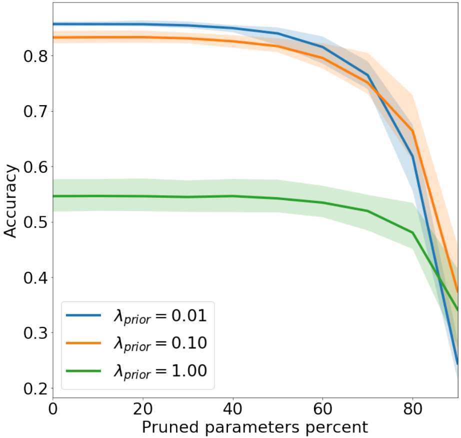
$\lambda_{\text{prior}} \ll \lambda_{\text{likelihood}}$

$\lambda_{\text{prior}} = \lambda_{\text{likelihood}}$

$\lambda_{\text{prior}} \gg \lambda_{\text{likelihood}}$

Example

λ_{prior} controls the importance of the prior distribution. With its increasing the model complexity decreases.



Grebenkova, Bakhteev, Strijov. Hypernetworks for deep model complexity control, 2021. (*work in progress*).

Current challenge

- Can we control the model complexity at the inference step?
- Can we select robust architrecture?
- Can we train not a single model, but an ensemble of models? Can we control their deiversity?

Model complexity control

Hypernetworks

A hypernetwork is a mapping from a set of variables responsible for the properties of a desired model to a set of its parameters.

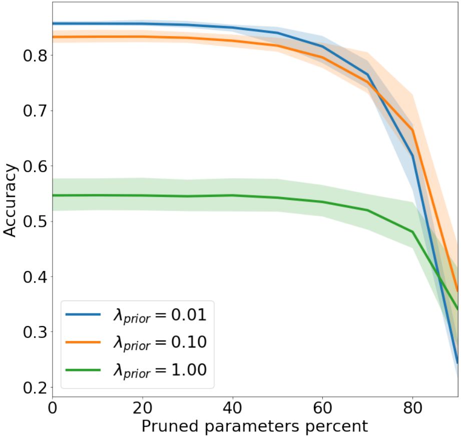
Optimize the model with hypernetworks in the following optimization procedure:

$$\mathbb{E}_{\lambda\sim P(\lambda)}(\log p(\mathfrak{D}|\mathrm{w}(\lambda)) - \lambda D_{\mathsf{KL}}(q(\mathrm{w}(\lambda))||p(\mathrm{w})) \to \max\,.$$

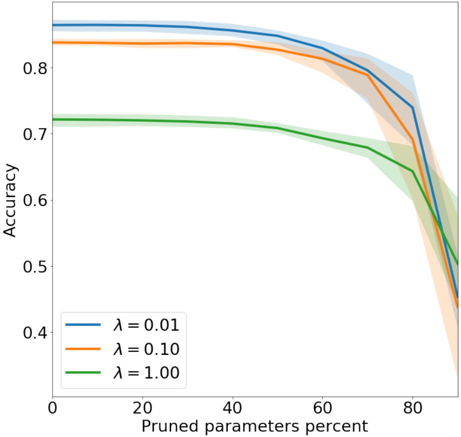
Theorem, Grebenkova 2021

The hypernetwork approximates not only deep learning model's performance, but also it's statistical propoerties.

Example



CNN



CNN with hypernetwork

Grebenkova, Bakhteev, Strijov. Hypernetworks for deep model complexity control, 2021. (*work in progress*).

Architecture complexity control

The hypernetworks can approximate not only the model parameter w, but also structural parameters $\boldsymbol{\gamma}$.

Baseline: DARTS

A model architecture is a directed graph with non-linear operations $\mathrm{f}^{(i,j)}$ that are induced by basic functions $\mathrm{g}^{(i,j)}$ with weights obtained by softmax function application:

$$\mathrm{f}^{(i,j)}(\mathrm{x}) = \langle \mathbf{softmax}(\boldsymbol{\gamma}^{(i,j)}), \mathrm{g}^{(i,j)}(\mathrm{x})\rangle$$

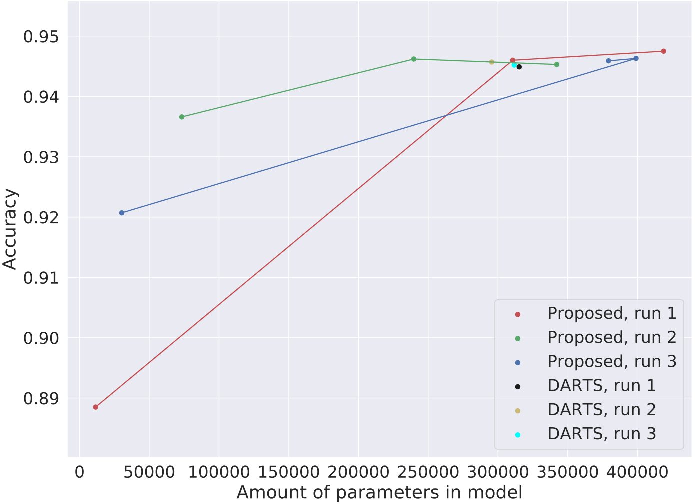
Our proposal

To use a mapping $\boldsymbol{\gamma}(\lambda_n)$ instead of constant structural parameters $\boldsymbol{\gamma}(\lambda_n)$, where λ_n is a regularization term for the loss function:

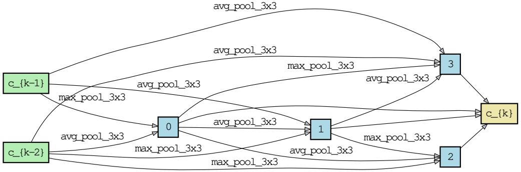
$$\mathrm{E}_{\lambda_n}\left(\log p(\mathrm{y}|\mathrm{X}, \mathrm{w}, \Gamma(\lambda_n)) + \lambda_n \sum_{(i,j)} \langle \mathbf{softmax}\left(\frac{\boldsymbol{\gamma}(\boldsymbol{\lambda_n})^{(i,j)}}{\lambda_{\mathbf{temp}}}\right), \mathrm{n}(\mathrm{g}^{(i,j)})\rangle\right),$$

where $\mathrm{n}(\mathrm{g}^{(i,j)})$ is a vector of amount of parameters for all the basic functions g.

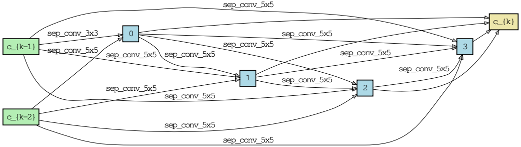
Example: Fashion-MNIST



Example



Simple CNN cell architecture



Complex CNN cell architeuctre

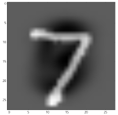
Yakovlev, Grebenkova, Bakhteev, Strijov. Automated architecture search with model complexity control, 2021. (*work in progress*).

Robustness and architecture ensembling

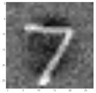
Intro + references

Experiments: MNIST

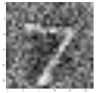
Noise adjusment $\mathcal{N}(0, \sigma^2 I)$:



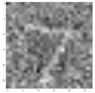
Original images



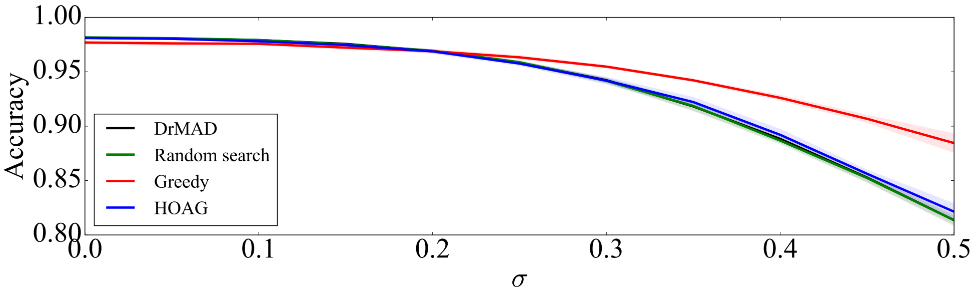
$\sigma = 0.1$



$\sigma = 0.25$



$\sigma = 0.5$



Plots on MNIST

Ensemblings

Plots on MNIST

References