Байесовская оптимизация на основе аппроксимации с помощью нейронной сети

Лисов Роман

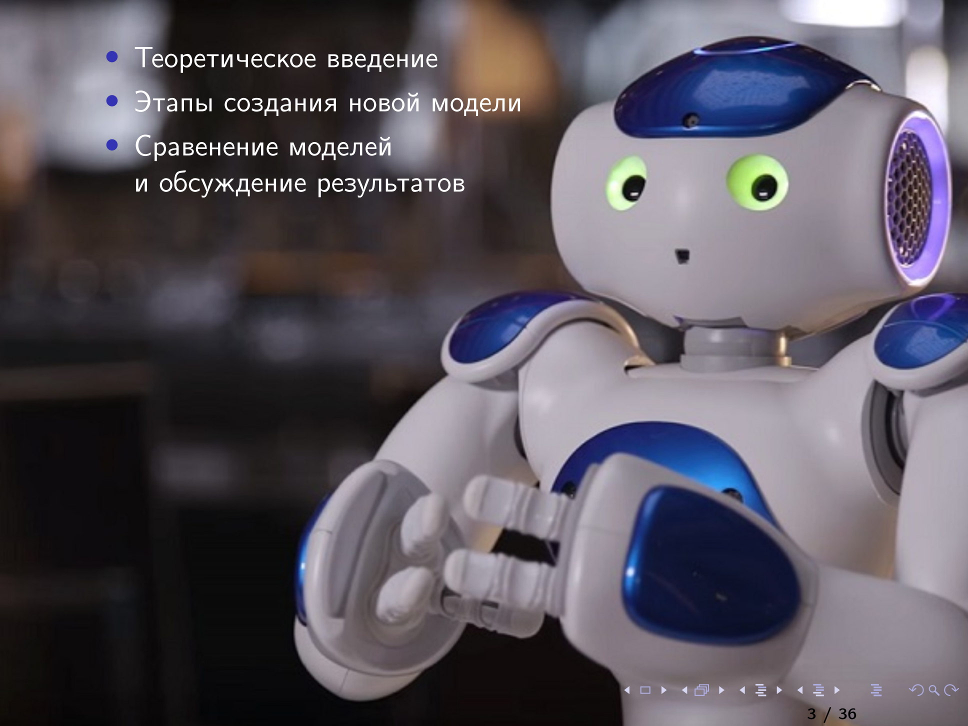
Московский физико-технический институт
(государственный университет)

10 мая 2019 г.

Постановка задачи

- Автор задачи — Максим Панов
- Цель работы — Построить модель байесовского оптимизатора на основе работы библотеки GPyOpt, аппроксимируя при этом функцию с помощью нейронной сети

- Теоретическое введение
- Этапы создания новой модели
- Сравенение моделей
  и обсуждение результатов



Определение: Гауссовский случайный процесс - это случайный процесс, все конечномерные распределения которого имеют нормальное распределение.

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})], \\ k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))] \end{aligned}$$

Обозначим:

$$f(\mathbf{x}) \sim \mathcal{GP}\big(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\big)$$

Частным случаем Гауссовского процесса может быть модель линейной регрессии (см. Backup), в которой input vectors проецируются в пространство $\phi(x)$, тогда

$$f(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^\top \mathbf{w},$$

Также следуя Байесовскому формализму:

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \Sigma_p)$$

И для модели Байесовской линейной регрессии получаем

$$\begin{aligned} \mathbb{E}[f(\mathbf{x})] &= \boldsymbol{\phi}(\mathbf{x})^\top \mathbb{E}[\mathbf{w}] = 0, \\ \mathbb{E}[f(\mathbf{x})f(\mathbf{x}')] &= \boldsymbol{\phi}(\mathbf{x})^\top \mathbb{E}[\mathbf{w}\mathbf{w}^\top]\boldsymbol{\phi}(\mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^\top \Sigma_p \boldsymbol{\phi}(\mathbf{x}'). \end{aligned}$$

Возьмём ковариационную функцию:

$$\operatorname{cov}\left(f(\mathbf{x}_p), f(\mathbf{x}_q)\right) = k(\mathbf{x}_p, \mathbf{x}_q) = \exp\left(-\tfrac{1}{2}|\mathbf{x}_p - \mathbf{x}_q|^2\right)$$

Таким образом предсказание модели можно обозначить:

$$\mathbf{f}_* \sim \mathcal{N}(\mathbf{0}, K(X_*, X_*))$$

Совместное распределение:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right)$$

Получаем предсказание в модели без шума:

$$\begin{aligned} \mathbf{f}_*|X_*, X, \mathbf{f} \sim \mathcal{N}(&K(X_*,X)K(X,X)^{-1}\mathbf{f}, \\ &K(X_*,X_*) - K(X_*,X)K(X,X)^{-1}K(X,X_*)) \end{aligned}$$

Если же добавить гауссовский шум:

$$\operatorname{cov}(y_p, y_q) = k(\mathbf{x}_p, \mathbf{x}_q) + \sigma_n^2\delta_{pq} \quad \text{or} \quad \operatorname{cov}(\mathbf{y}) = K(X,X) + \sigma_n^2 I$$

Окончательно

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X)+\sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right)$$

Предсказание принимает вид:

$$\begin{aligned} \mathbf{f}_*|X,\mathbf{y},X_* &\sim \mathcal{N}\big(\bar{\mathbf{f}}_*,\ \mathrm{cov}(\mathbf{f}_*)\big), \quad \text{where} \\ \bar{\mathbf{f}}_* &\triangleq \mathbb{E}[\mathbf{f}_*|X,\mathbf{y},X_*] = K(X_*,X)[K(X,X)+\sigma_n^2 I]^{-1}\mathbf{y}, \\ \mathrm{cov}(\mathbf{f}_*) &= K(X_*,X_*) - K(X_*,X)[K(X,X)+\sigma_n^2 I]^{-1}K(X,X_*) \end{aligned}$$

Чтобы оценить сложность настройки параметров

$$p(\mathbf{y}|X) = \int p(\mathbf{y}|\mathbf{f},X)p(\mathbf{f}|X)\,d\mathbf{f}$$

Имеем:

$$\log p(\mathbf{y}|X) = -\tfrac{1}{2}\mathbf{y}^\top(K+\sigma_n^2 I)^{-1}\mathbf{y} - \tfrac{1}{2}\log|K+\sigma_n^2 I| - \tfrac{n}{2}\log 2\pi$$

Суррогатная оптимизация - оптимизация дорогой для вычисления функции с помощью другой, аппроксимирующей её.

- Задача: оптимизация функции $f(x)$
- Если функция дорогая для вычисления, строим аппроксимацию $\widehat{f}(x)$
- Оптимизируем полученную $\widehat{f}(x)$
- Найдя оптимум для аппроксимации, с большой вероятностью найдём оптимум и для исходной $f(x)$

Зная среднее значение $\widehat{f}(x)$ и ошибку предсказания в каждой точке, будем составлять критерий выбора следующих точек для вычисления - infill criteria

Два главных направления оптимизации:

- exploration - поиск min в районах с малым средним $\widehat{y}(x)$
- eploitation - где большая ошибка предсказания $s(x)$

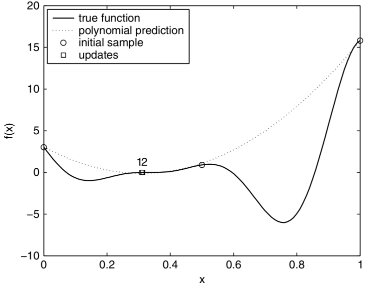
Aqusition functions:

- Probability of improvement
- Statistical Lower bounds
- Энтропийный поиск
- Expected improvement

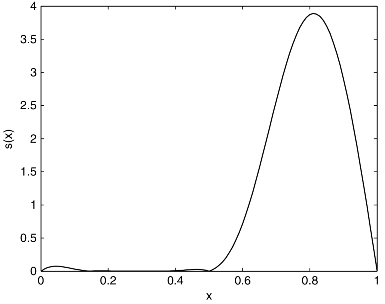
Формула для EI:

$$E[I(\mathbf{x})] = \begin{cases} (y_{\min} - \widehat{y}(\mathbf{x}))\Phi\left(\frac{y_{\min} - \widehat{y}(\mathbf{x})}{\widehat{s}(\mathbf{x})}\right) + s\phi\left(\frac{y_{\min} - \widehat{y}(\mathbf{x})}{\widehat{s}(\mathbf{x})}\right) & \text{if } s > 0 \\ 0 & \text{if } s = 0 \end{cases}$$

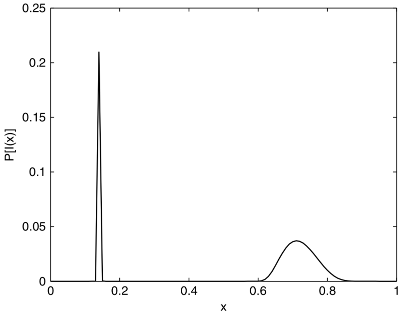
Example of a function to optimize:



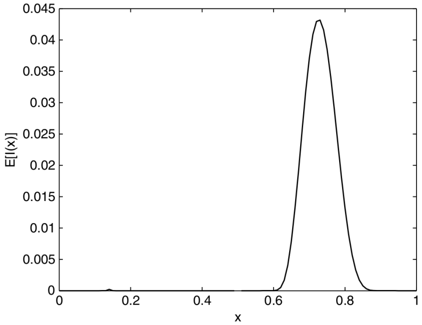
Mean squared error:



Probability of improvement



Expected improvement



Идея: Оптимизиация на основе гауссовских процессов существенно усложняется с ростом размерности и объёма выборки:
при обучении и предсказании асимптотическая сложность вычислений модели составляет $O(n^3)$.

Обучение:

$$\mathbf{f}_*|X,\mathbf{y},X_* \sim \mathcal{N}\big(\bar{\mathbf{f}}_*,\ \mathrm{cov}(\mathbf{f}_*)\big), \quad \text{where}$$
$$\bar{\mathbf{f}}_* \triangleq \mathbb{E}[\mathbf{f}_*|X,\mathbf{y},X_*] = K(X_*,X)[K(X,X)+\sigma_n^2 I]^{-1}\mathbf{y},$$
$$\mathrm{cov}(\mathbf{f}_*) = K(X_*,X_*) - K(X_*,X)[K(X,X)+\sigma_n^2 I]^{-1}K(X,X_*)$$

Предсказание:

$$\log p(\mathbf{y}|X) = -\tfrac{1}{2}\mathbf{y}^\top(K+\sigma_n^2 I)^{-1}\mathbf{y} - \tfrac{1}{2}\log|K+\sigma_n^2 I| - \tfrac{n}{2}\log 2\pi$$

Этапы оптимизации:

- Составление дискретной сетки
- Обучение NN
- Применение метода Dropout к NN, получаем поточечные standard deviation и prediction
- Строим байесовский оптимизатор на основе библиотеки GPyOpt, при этом вместо высчитывания дисперсия и среднего значения функции с помощью ГП будем использовать данные выхода сетки

Dropout - метод борьбы с переобучением нейронных сетей.

Схема обычной NN:

$$\begin{aligned} z_i^{(l+1)} &= \mathbf{w}_i^{(l+1)}\mathbf{y}^l + b_i^{(l+1)}, \\ y_i^{(l+1)} &= f(z_i^{(l+1)}), \end{aligned}$$

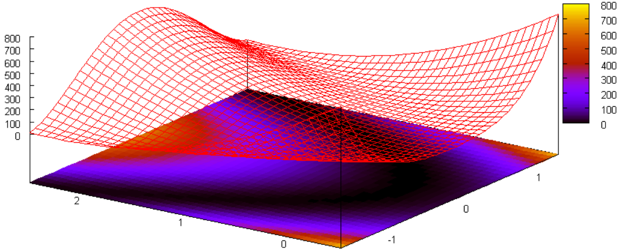
Схема NN с применением Dropout:

$$\begin{aligned} r_j^{(l)} &\sim \mathrm{Bernoulli}(p), \\ \widetilde{\mathbf{y}}^{(l)} &= \mathbf{r}^{(l)} * \mathbf{y}^{(l)}, \\ z_i^{(l+1)} &= \mathbf{w}_i^{(l+1)}\widetilde{\mathbf{y}}^l + b_i^{(l+1)}, \\ y_i^{(l+1)} &= f(z_i^{(l+1)}). \end{aligned}$$

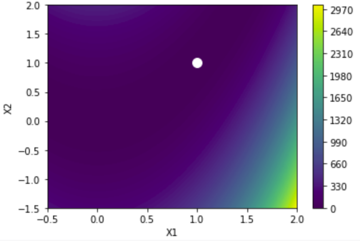
Примеры:

- двумерная функция *Rosenbrock*
- многомерная функция *Rosenbrock8d*
- функция *Branin*

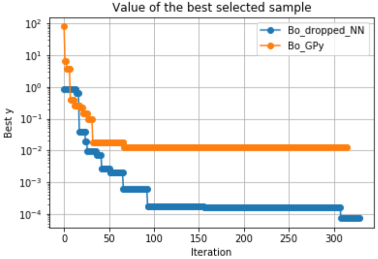
Функция розенброка для двумерноего случая:



Rosenbrock

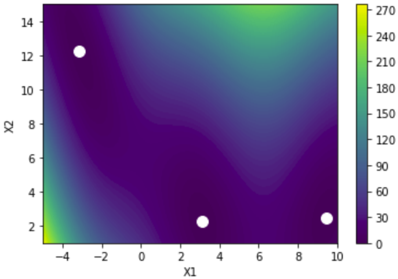


Сходимость за 25 мин

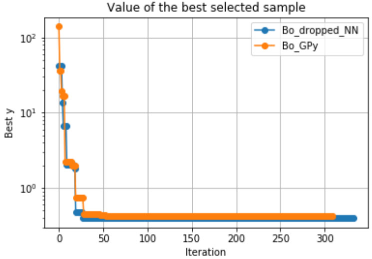


	x_{min}	f_{min}
Глобальный минимум	(1.00, 1.00)	0
BO-dropout-NN	(1.01, 1.02)	7e-05
BO-GPyOpt	(1.10, 1.23)	1e-02

Branin

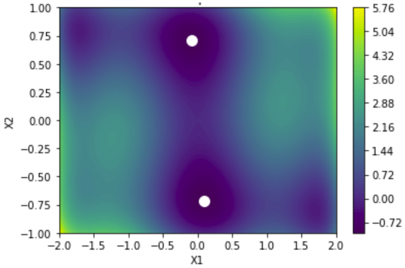


Сходимость за 25 мин

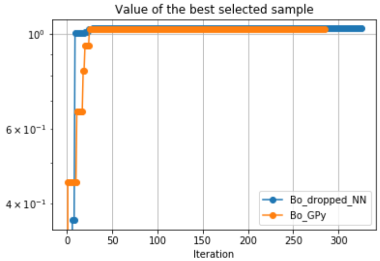


	x_{min}	f_{min}
Глобальный минимум	(9.40, 2.50)	0.397
BO-dropout-NN	(-3.14, 12.28)	0.398
BO-GPyOpt	(3.07, 2.32)	0.421

Six-hump camel

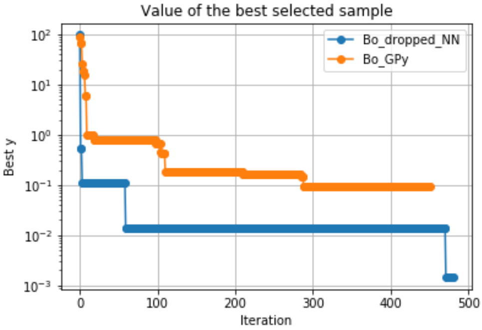


Сходимость за 25 мин



	x_{min}	f_{min}
Глобальный минимум	(0.1, -0.71)	-1.0316
BO-dropout-NN	(0.09, -0.71)	-1.0316
BO-GPyOpt	(0.12, -0.71)	-1.0286

Сходимость за 2ч с keep-prob = 0.5



Результаты оптимизации

keep-prob	f_{min}
0.50	0.00145
0.65	0.00156
0.75	0.01941
0.85	0.01603

	x_{min}	f_{min}
BO-dropout-NN	(0.98 0.96 1.21 1.95 1.83 1.93 1.11 0.5)	0.00145
BO-GPyOpt	(0.70 0.49 1.36 0.91 2.00 1.67 1.13 2.0)	0.09253

- Видим, что на функциях размерности 2dim новый алгоритм BO-dropout-NN работает быстрее и точнее стандартного алгоритма BO-GPyOpt
- для *Rosenbrock8d* за 120 минут работы BO-dropout-NN делает больше итераций оптимизации и точнее находит минимум функции.
- Проект можно развивать дальше, начав применять полученный положительный результат в областях использования Байесовской оптимизации.
- Ссылка на программный код: https://github.com/lisovrv/IITP

Спасибо за внимание!

Training set:

$$\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid i = 1, \ldots, n\}$$

Байесовский подход к линейной регрессии:

$$f(\mathbf{x}) = \mathbf{x}^\top \mathbf{w}, \qquad y = f(\mathbf{x}) + \varepsilon$$

с шумом:

$$\varepsilon \sim \mathcal{N}(0,\ \sigma_n^2).$$

Функция правдоподобия:

$$p(\mathbf{y}|X, \mathbf{w}) = \prod_{i=1}^{n} p(y_i|\mathbf{x}_i, \mathbf{w}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left(-\frac{(y_i - \mathbf{x}_i^\top \mathbf{w})^2}{2\sigma_n^2}\right)$$
$$= \frac{1}{(2\pi\sigma_n^2)^{n/2}} \exp\left(-\frac{1}{2\sigma_n^2}|\mathbf{y} - X^\top \mathbf{w}|^2\right) = \mathcal{N}(X^\top \mathbf{w}, \sigma_n^2 I)$$

Следуя Байесовскому формализму:

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0},\ \Sigma_p)$$

По формуле Байеса:

$$\text{posterior} \; = \; \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}, \qquad p(\mathbf{w}|\mathbf{y}, X) \; = \; \frac{p(\mathbf{y}|X, \mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|X)}$$

где marginal likelihood -

$$p(\mathbf{y}|X) \; = \; \int p(\mathbf{y}|X, \mathbf{w})p(\mathbf{w})\, d\mathbf{w}$$

Апостариорное распределение

$$p(\mathbf{w}|X, \mathbf{y}) \; \sim \; \mathcal{N}(\bar{\mathbf{w}} = \frac{1}{\sigma_n^2} A^{-1} X\mathbf{y}, \; A^{-1})$$

где матрица A равна:

$$A = \sigma_n^{-2} X X^\top + \Sigma_p^{-1}$$

Предсказание линейной модели:

$$p(f_*|\mathbf{x}_*, X, \mathbf{y}) = \int p(f_*|\mathbf{x}_*, \mathbf{w})p(\mathbf{w}|X, \mathbf{y})\, d\mathbf{w}$$
$$= \mathcal{N}\left(\frac{1}{\sigma_n^2}\mathbf{x}_*^\top A^{-1} X\mathbf{y},\ \mathbf{x}_*^\top A^{-1}\mathbf{x}_*\right)$$

Спроецируем input vectors в пространство $\phi(x)$ - Feature Space

$$\boldsymbol{\phi}(x) = (1, x, x^2, x^3, \ldots)^\top$$

Тогда

$$f(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^\top \mathbf{w},$$

и предсказание модели выражается как

$$f_*|\mathbf{x}_*, X, \mathbf{y} \sim \mathcal{N}\left(\frac{1}{\sigma_n^2}\boldsymbol{\phi}(\mathbf{x}_*)^\top A^{-1}\Phi\mathbf{y},\ \boldsymbol{\phi}(\mathbf{x}_*)^\top A^{-1}\boldsymbol{\phi}(\mathbf{x}_*)\right)$$

$$A = \sigma_n^{-2}\Phi\Phi^\top + \Sigma_p^{-1}$$

Перепишем в виде

$$f_*|\mathbf{x}_*, X, \mathbf{y} \sim \mathcal{N}\big(\boldsymbol{\phi}_*^\top \Sigma_p \Phi (K + \sigma_n^2 I)^{-1}\mathbf{y},\\ \boldsymbol{\phi}_*^\top \Sigma_p \boldsymbol{\phi}_* - \boldsymbol{\phi}_*^\top \Sigma_p \Phi (K + \sigma_n^2 I)^{-1} \Phi^\top \Sigma_p \boldsymbol{\phi}_*\big)$$

Заметим, что

$$k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^\top \Sigma_p \boldsymbol{\phi}(\mathbf{x}')$$

Оно же представимо

$$k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\psi}(\mathbf{x}) \cdot \boldsymbol{\psi}(\mathbf{x}')$$

- Kernel trick

Определение: Гауссовский случайный процесс - случайный процесс, все конечномерные распределения которого имеют нормальное распределение.

$$
\begin{aligned}
m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})], \\
k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]
\end{aligned}
$$

Обозначим:

$$f(\mathbf{x}) \sim \mathcal{GP}\big(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\big)$$

Для модели Байесовской линейной регрессии

$$
\begin{aligned}
\mathbb{E}[f(\mathbf{x})] &= \phi(\mathbf{x})^\top \mathbb{E}[\mathbf{w}] = 0, \\
\mathbb{E}[f(\mathbf{x}) f(\mathbf{x}')] &= \phi(\mathbf{x})^\top \mathbb{E}[\mathbf{w}\mathbf{w}^\top] \phi(\mathbf{x}') = \phi(\mathbf{x})^\top \Sigma_p \phi(\mathbf{x}')
\end{aligned}
$$

Возьмём ковариационную функцию:

$$\operatorname{cov}\big(f(\mathbf{x}_p), f(\mathbf{x}_q)\big) = k(\mathbf{x}_p, \mathbf{x}_q) = \exp\big(-\tfrac{1}{2}|\mathbf{x}_p - \mathbf{x}_q|^2\big)$$

Таким образом предсказание модели можно обозначить:

$$\mathbf{f}_* \sim \mathcal{N}(\mathbf{0}, K(X_*, X_*))$$

Совместное распределение:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X) & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right)$$

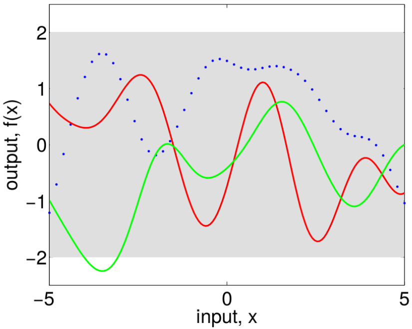
Получаем предсказание в модели без шума:

$$\begin{aligned} \mathbf{f}_*|X_*, X, \mathbf{f} \sim \mathcal{N}\big(&K(X_*,X)K(X,X)^{-1}\mathbf{f}, \\ &K(X_*,X_*) - K(X_*,X)K(X,X)^{-1}K(X,X_*)\big) \end{aligned}$$

Если же добавить гауссовский шум:

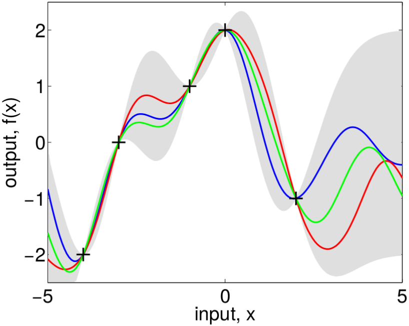
$$\operatorname{cov}(y_p, y_q) = k(\mathbf{x}_p, \mathbf{x}_q) + \sigma_n^2\delta_{pq} \quad \text{or} \quad \operatorname{cov}(\mathbf{y}) = K(X,X) + \sigma_n^2 I$$

Prior distribution on picture (a):



(a), prior

Posterior distribution on picture (b):



(b), posterior

Окончательно

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(X,X)+\sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right)$$

Предсказание принимает вид:

$$\begin{aligned} \mathbf{f}_*|X,\mathbf{y},X_* &\sim \mathcal{N}(\bar{\mathbf{f}}_*,\ \mathrm{cov}(\mathbf{f}_*)), \quad \text{where} \\ \bar{\mathbf{f}}_* &\triangleq \mathbb{E}[\mathbf{f}_*|X,\mathbf{y},X_*] = K(X_*,X)[K(X,X)+\sigma_n^2 I]^{-1}\mathbf{y}, \\ \mathrm{cov}(\mathbf{f}_*) &= K(X_*,X_*) - K(X_*,X)[K(X,X)+\sigma_n^2 I]^{-1}K(X,X_*) \end{aligned}$$

Чтобы оценить сложность настройки параметров

$$p(\mathbf{y}|X) = \int p(\mathbf{y}|\mathbf{f},X)p(\mathbf{f}|X)\,d\mathbf{f}$$

Имеем:

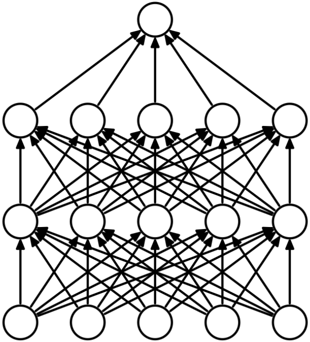
$$\log p(\mathbf{y}|X) = -\tfrac{1}{2}\mathbf{y}^\top(K+\sigma_n^2 I)^{-1}\mathbf{y} - \tfrac{1}{2}\log|K+\sigma_n^2 I| - \tfrac{n}{2}\log 2\pi$$

Dropout - метод борьбы с переобучением нейронных сетей.
Схема обычной NN:

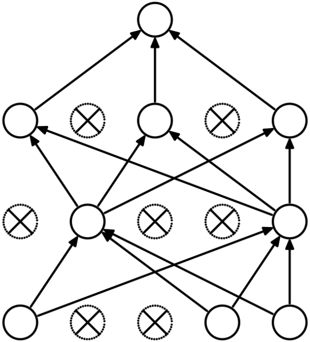
$$\begin{aligned} z_i^{(l+1)} &= \mathbf{w}_i^{(l+1)}\mathbf{y}^l + b_i^{(l+1)}, \\ y_i^{(l+1)} &= f(z_i^{(l+1)}), \end{aligned}$$

Схема NN с применением Dropout:

$$\begin{aligned} r_j^{(l)} &\sim \text{Bernoulli}(p), \\ \widetilde{\mathbf{y}}^{(l)} &= \mathbf{r}^{(l)} * \mathbf{y}^{(l)}, \\ z_i^{(l+1)} &= \mathbf{w}_i^{(l+1)}\widetilde{\mathbf{y}}^l + b_i^{(l+1)}, \\ y_i^{(l+1)} &= f(z_i^{(l+1)}). \end{aligned}$$



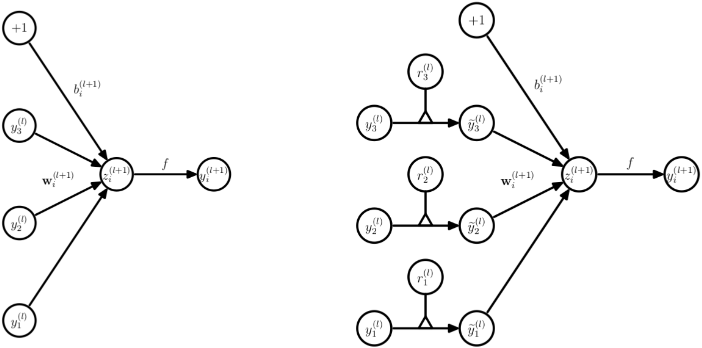
(a) Standard Neural Net



(b) After applying dropout.

Backup

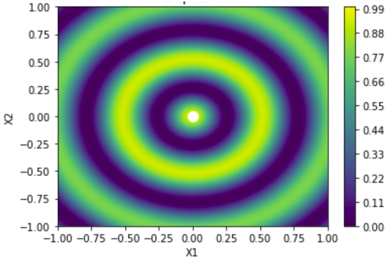
DropOut



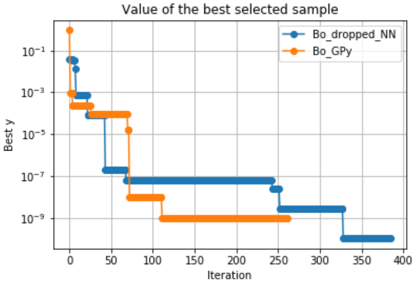
(a) Standard network

(b) Dropout network

Dropwave

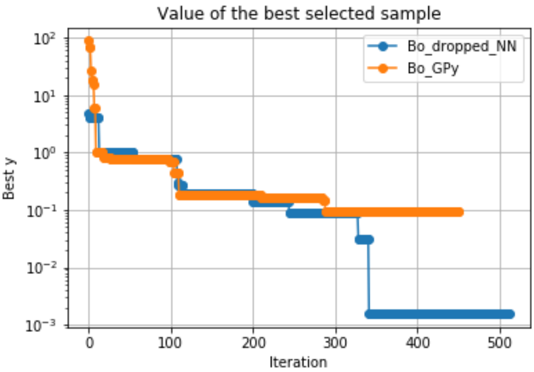


Сходимость за 25 мин



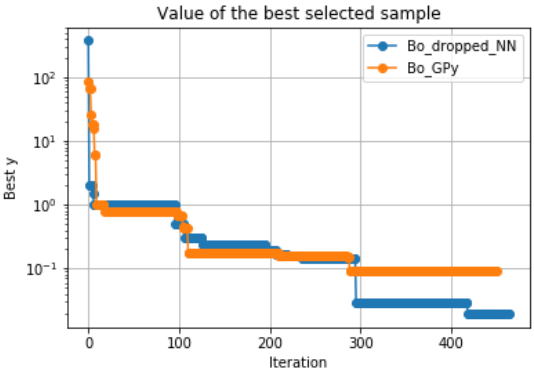
	x_{min}	f_{min}
Глобальный минимум	(0, 0)	0
BO-dropout-NN	(0.75, 0.23)	1e-10
BO-GPyOpt	(0.67, -0.41)	1e-09

Сходимость за 2 часа с keep-probability = 0.65



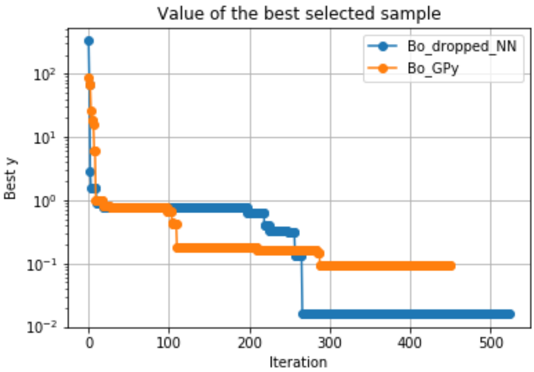
	f_{min}
BO-dropout-NN	0.00156
BO-GPyOpt	0.09253

Сходимость за 2 часа с keep-probability $= 0.75$



	f_{min}
BO-dropout-NN	0.01941
BO-GPyOpt	0.09253

Сходимость за 2 часа с keep-probability = 0.85



	f_{min}
BO-dropout-NN	0.01603
BO-GPyOpt	0.09253