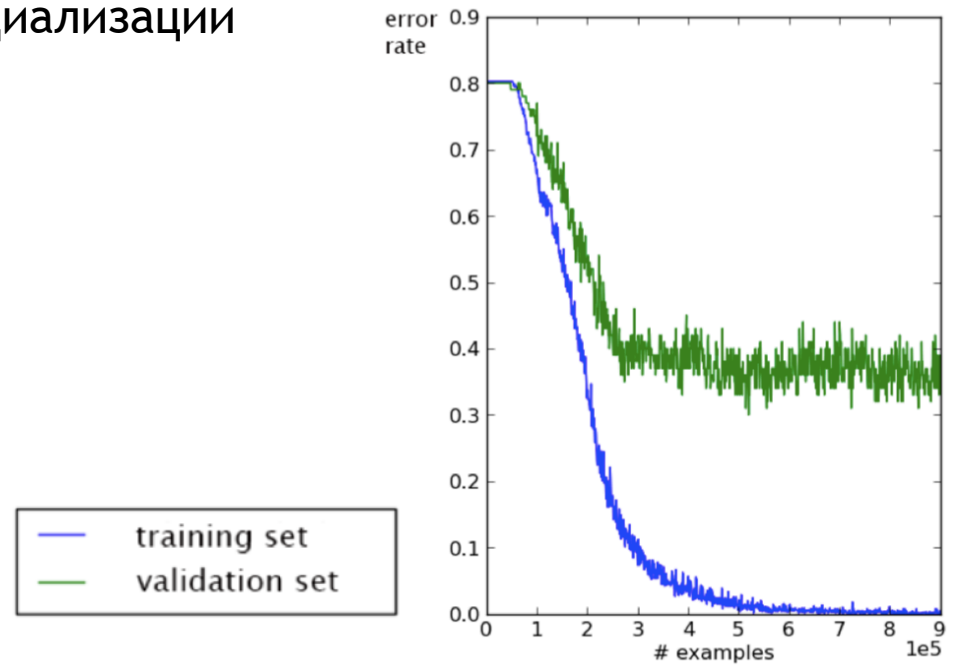
Ансамблирование нейронных сетей

Гальцев Даниил
НИУ ВШЭ
15.11.2019

Ансамбли

Ансамбль – композиция нескольких моделей для решения задачи машинного обучения

- Ансамбли позволяют уменьшить разброс предсказаний и уменьшить переобучение
- Нейронные сети склонны к переобучению и чувствительны к обучающим данным и параметрам инициализации



Model Averaging

Как обучать разные модели:

- Получить выборки бутстрепом
- Случайно инициализировать начальные параметры сети и перемешать обучающую выборку

Объединяем результаты нескольких моделей:

- Для регрессии – усредняем полученные значение
- Для классификации – суммируем предсказания и выбираем класс с наибольшей вероятностью

Stacked Generalization

Вместо усреднения результатов обучим новую модель, объединяющую ответы.

Обучение на одной выборке может привести к переобучению. Решения:

- Можно выделить из тренировочной выборки валидационную выборку
- Можно подготовить данные для объединяющей модели с помощью кросс-валидации, а обучать модели по отдельности.

Horizontal Voting Ensemble

- Обучение нескольких моделей может занимать очень много времени
- Для получения ансамбля можно брать веса модели в различные моменты обучения
- В Horizontal Voting Ensemble усредняются веса модели в нескольких последних эпохах

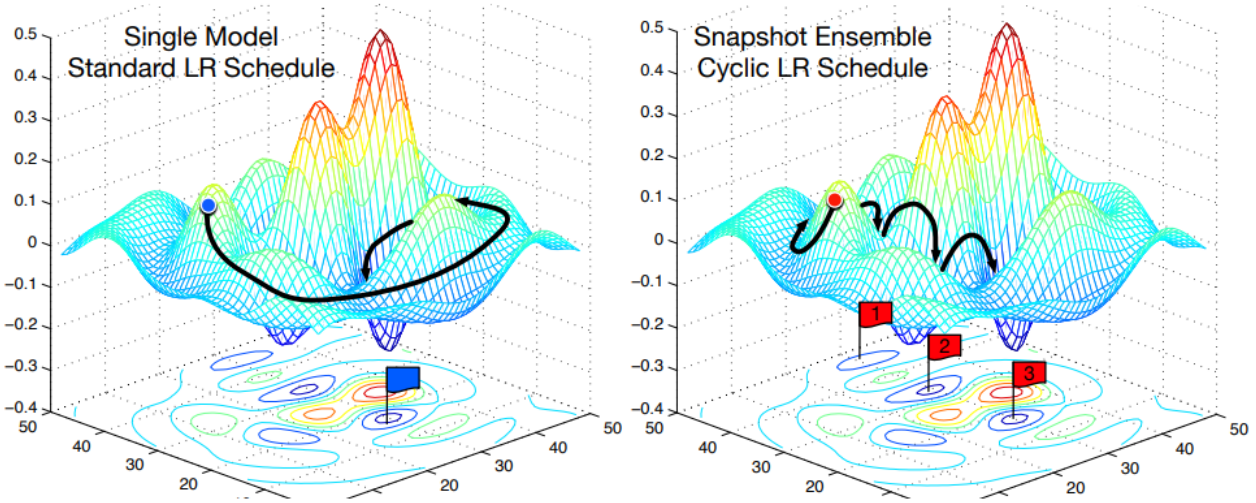
Horizontal Voting Ensemble

- Тестирование на данных из ICML 2013 Black Box Challenge
- Модель 2 – глубокая нейронная сеть с нейронами 1875-1500-1000-1500-1200-1500- 1500-1500-1500-9
- Модель 4 – ансамбль модели 2 с 651 по 850 эпоху

	MODEL 2	MODEL 4
ACCURACY(PUBLIC TEST SET)	0.66660	0.68220
ACCURACY(PRIVATE TEST SET)	0.65120	0.67240

Snapshot Ensemble

- Ансамбли лучше работают, когда предсказания моделей не коррелируют
- При обучении нейронной сети есть много локальных минимумов, которые могут содержат полезную информацию



Snapshot Ensemble

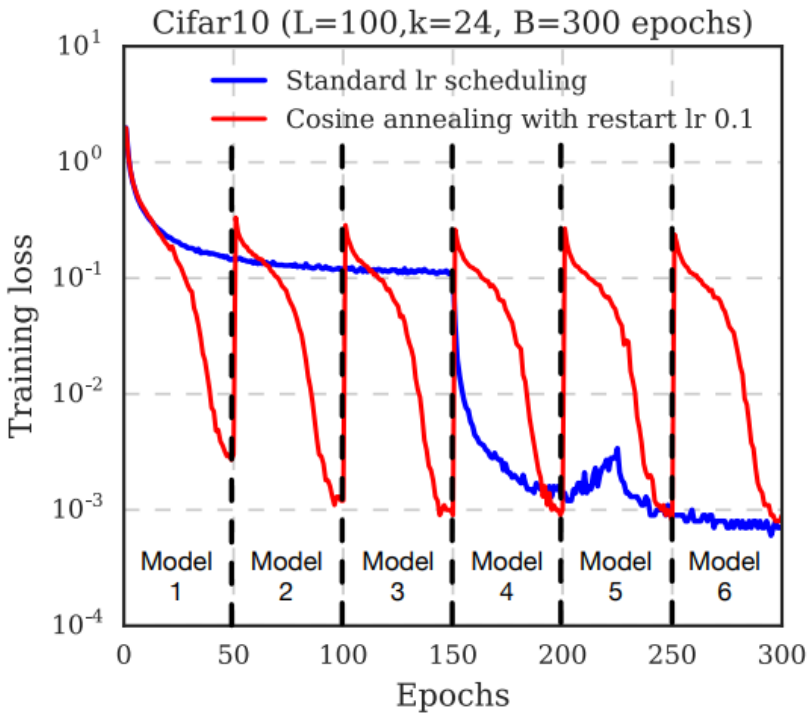
- Для схождения будем быстро уменьшать темп обучения
- Для получения различных локальных минимумов воспользуемся циклическим темпом обучения

$$\alpha(t) = f\left(\mathrm{mod}\left(t-1, \lceil T/M \rceil\right)\right)$$

$$\alpha(t) = \frac{\alpha_0}{2}\left(\cos\left(\frac{\pi \mathrm{mod}(t-1, \lceil T/M \rceil)}{\lceil T/M \rceil}\right) + 1\right)$$

- Последние *m* моделей объединяем в ансамбль

$$h_{\text{Ensemble}} = \frac{1}{m}\sum_{0}^{m-1} h_{M-i}(\mathbf{x})$$



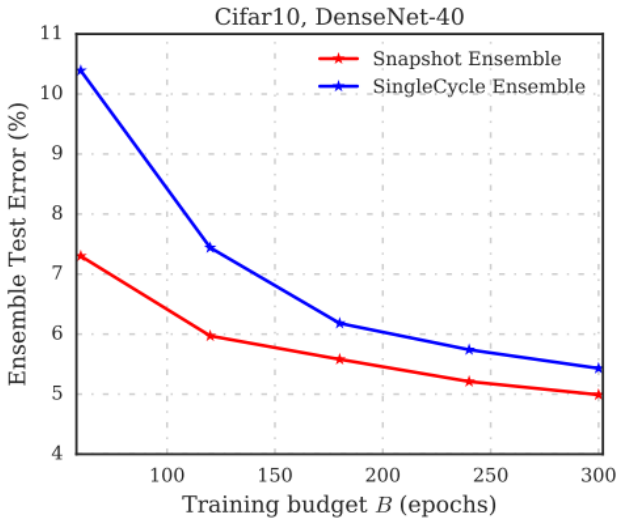
Snapshot Ensemble

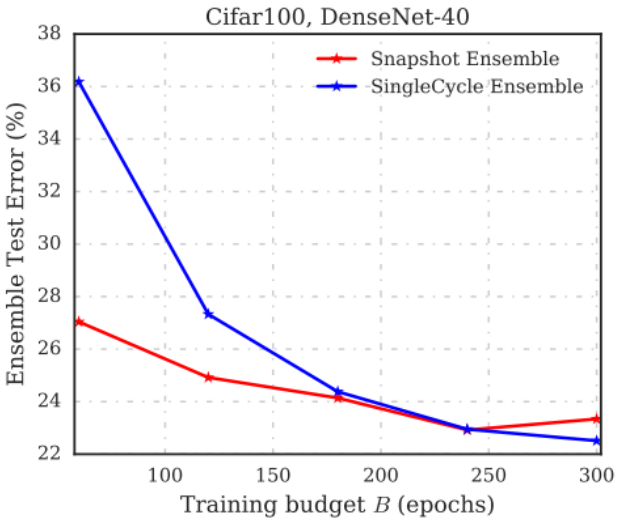
Тестовая ошибка в экспериментах

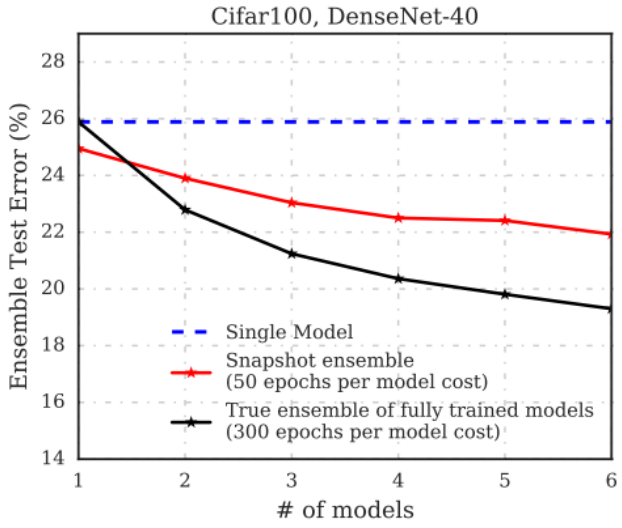
	Method	C10	C100	SVHN	Tiny ImageNet
ResNet-110	Single model	5.52	28.02	1.96	46.50
	NoCycle Snapshot Ensemble	5.49	26.97	1.78	43.69
	SingleCycle Ensembles	6.66	24.54	1.74	42.60
	Snapshot Ensemble ($\alpha_0 = 0.1$)	5.73	25.55	**1.63**	40.54
	Snapshot Ensemble ($\alpha_0 = 0.2$)	**5.32**	**24.19**	1.66	**39.40**
Wide-ResNet-32	Single model	5.43	23.55	1.90	39.63
	Dropout	4.68	22.82	1.81	36.58
	NoCycle Snapshot Ensemble	5.18	22.81	1.81	38.64
	SingleCycle Ensembles	5.95	21.38	1.65	35.53
	Snapshot Ensemble ($\alpha_0 = 0.1$)	**4.41**	**21.26**	1.64	35.45
	Snapshot Ensemble ($\alpha_0 = 0.2$)	4.73	21.56	**1.51**	**32.90**
DenseNet-40	Single model	5.24*	24.42*	1.77	39.09
	Dropout	6.08	25.79	1.79*	39.68
	NoCycle Snapshot Ensemble	5.20	24.63	1.80	38.51
	SingleCycle Ensembles	5.43	22.51	1.87	38.00
	Snapshot Ensemble ($\alpha_0 = 0.1$)	4.99	23.34	**1.64**	37.25
	Snapshot Ensemble ($\alpha_0 = 0.2$)	**4.84**	**21.93**	1.73	**36.61**
DenseNet-100	Single model	3.74*	19.25*	-	-
	Dropout	3.65	18.77	-	-
	NoCycle Snapshot Ensemble	3.80	19.30	-	-
	SingleCycle Ensembles	4.52	18.38		
	Snapshot Ensemble ($\alpha_0 = 0.1$)	3.57	18.12	-	-
	Snapshot Ensemble ($\alpha_0 = 0.2$)	**3.44**	**17.41**	-	-

Snapshot Ensemble

Зависимость тестовой ошибки от время обучения и сравнение с обычным ансабмлем

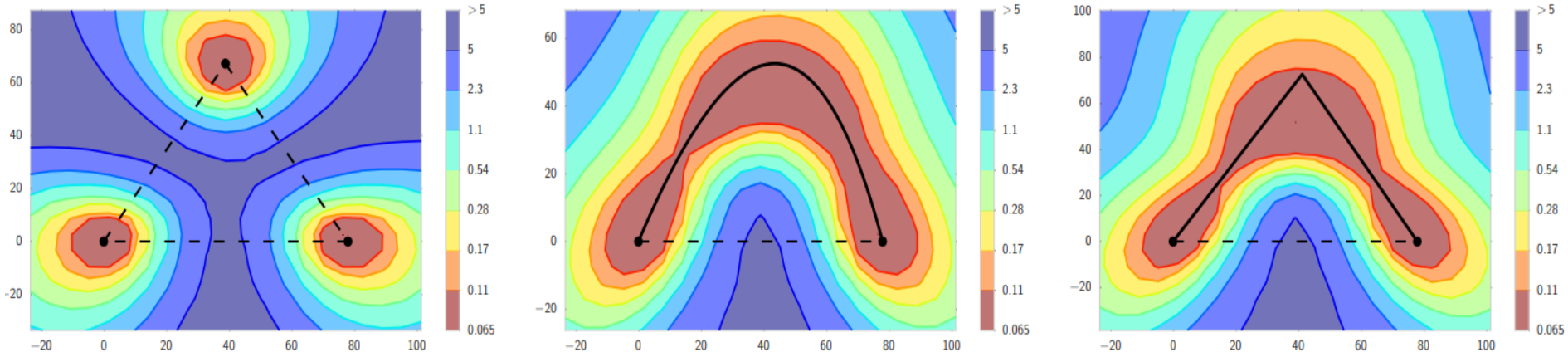






Fast Geometric Ensembling

- Snapshot ensemble берет модели из окрестностей различных локальных минимумов
- Часто существуют общие области низких значений функции потерь

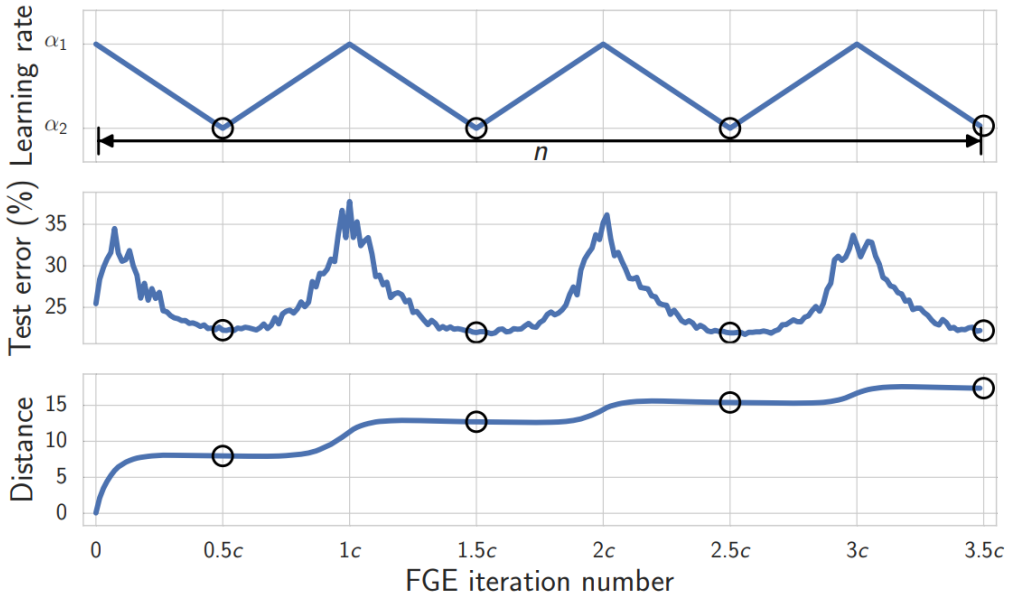


FGE

- Начнем обучение с предобученной нейронной сети
- Воспользуемся быстрым (2-4 эпохи) циклическим темпом обучения

$$\alpha(i) = \begin{cases} (1 - 2t(i))\alpha_1 + 2t(i)\alpha_2 & 0 < t(i) \leq \frac{1}{2} \\ (2 - 2t(i))\alpha_2 + (2t(i) - 1)\alpha_1 & \frac{1}{2} < t(i) \leq 1 \end{cases} \qquad t(i) = \frac{1}{c}(\mathrm{mod}(i - 1, c) + 1)$$

- Берем модели, когда темп обучения минимален



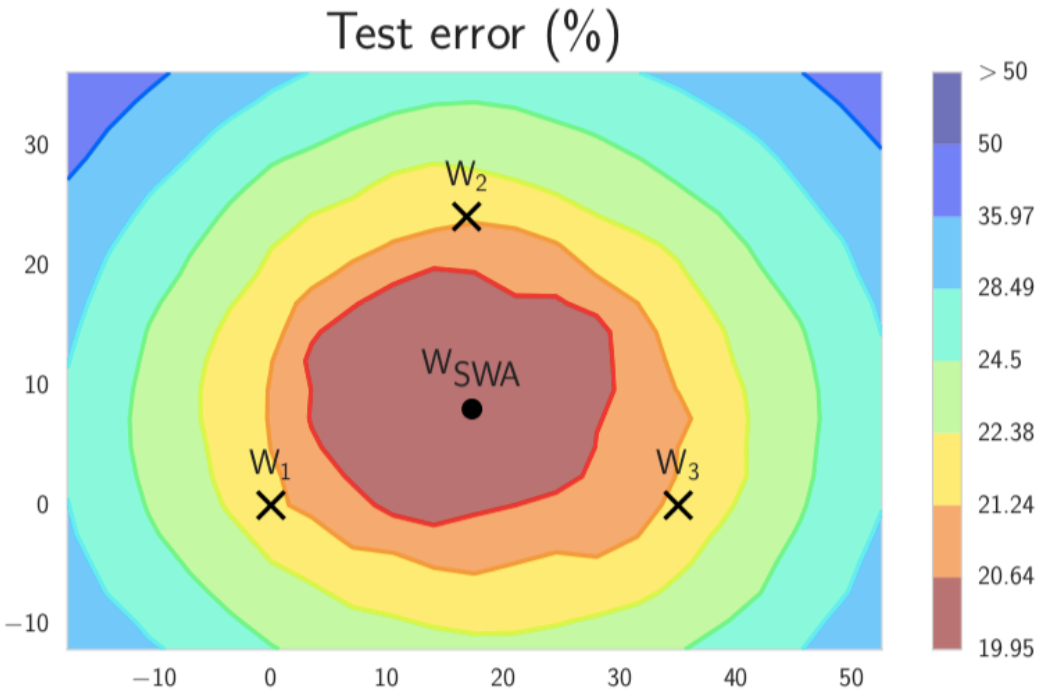
FGE

Результаты экспериментов

DNN (Budget)	method	CIFAR-100			CIFAR-10		
		$1B$	$2B$	$3B$	$1B$	$2B$	$3B$
VGG-16 (200)	Ind	27.4 ± 0.1	25.28	24.45	6.75 ± 0.16	5.89	5.9
	SSE	26.4 ± 0.1	25.16	24.69	6.57 ± 0.12	6.19	5.95
	FGE	$\mathbf{25.7 \pm 0.1}$	**24.11**	**23.54**	$\mathbf{6.48 \pm 0.09}$	**5.82**	**5.66**
ResNet-164 (150)	Ind	21.5 ± 0.4	19.04	18.59	4.72 ± 0.1	**4.1**	**3.77**
	SSE	20.9 ± 0.2	19.28	18.91	4.66 ± 0.02	4.37	4.3
	FGE	$\mathbf{20.2 \pm 0.1}$	**18.67**	**18.21**	$\mathbf{4.54 \pm 0.05}$	4.21	3.98
WRN-28-10 (200)	Ind	19.2 ± 0.2	17.48	17.01	3.82 ± 0.1	3.4	**3.31**
	SSE	17.9 ± 0.2	17.3	16.97	3.73 ± 0.04	3.54	3.55
	FGE	$\mathbf{17.7 \pm 0.2}$	**16.95**	**16.88**	$\mathbf{3.65 \pm 0.1}$	**3.38**	3.52

Stochastic Weight Averaging

- Использование ансамблей приводит к увеличению времени на предсказание
- Попробуем приблизить ансамбль с помощью усреднения весов

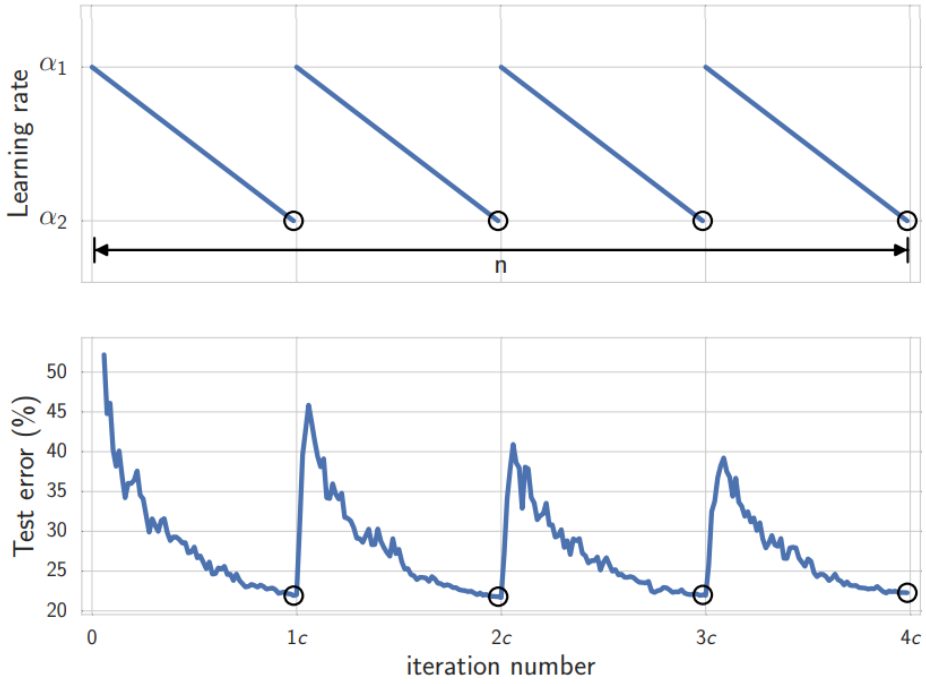


SWA

- Начнем обучение с предобученной нейронной сети
- Воспользуемся быстрым (2-4 эпохи) циклическим темпом обучения

$$\alpha(i) = (1 - t(i))\alpha_1 + t(i)\alpha_2 \qquad t(i) = \frac{1}{c}\left(\mathrm{mod}(i-1, c) + 1\right)$$

- Берем модели, когда темп обучения минимален
- Веса итоговых моделей усредняем



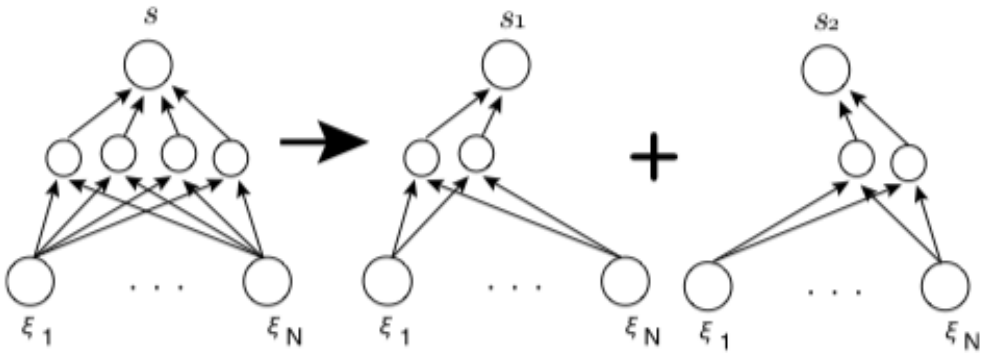
SWA

Результаты экспериментов

DNN (Budget)	SGD	FGE (1 Budget)	SWA		
			1 Budget	1.25 Budgets	1.5 Budgets
CIFAR-100					
VGG-16 (200)	72.55 ± 0.10	74.26	73.91 ± 0.12	74.17 ± 0.15	74.27 ± 0.25
ResNet-164 (150)	78.49 ± 0.36	79.84	79.77 ± 0.17	80.18 ± 0.23	80.35 ± 0.16
WRN-28-10 (200)	80.82 ± 0.23	82.27	81.46 ± 0.23	81.91 ± 0.27	82.15 ± 0.27
PyramidNet-272 (300)	83.41 ± 0.21	–	–	83.93 ± 0.18	84.16 ± 0.15
CIFAR-10					
VGG-16 (200)	93.25 ± 0.16	93.52	93.59 ± 0.16	93.70 ± 0.22	93.64 ± 0.18
ResNet-164 (150)	95.28 ± 0.10	95.45	95.56 ± 0.11	95.77 ± 0.04	95.83 ± 0.03
WRN-28-10 (200)	96.18 ± 0.11	96.36	96.45 ± 0.11	96.64 ± 0.08	96.79 ± 0.05
ShakeShake-2x64d (1800)	96.93 ± 0.10	–	–	97.16 ± 0.10	97.12 ± 0.06

Dropout

- Dropout используется для предотвращения переобучения
- Во время обучения не используется часть нейронов
- Во время предсказания используются все нейроны, но взвешенно

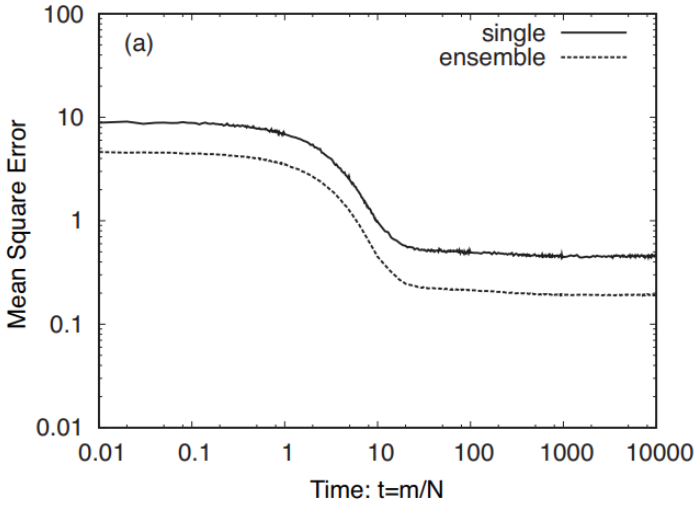


Сеть, разделенная для ансамбального обучения

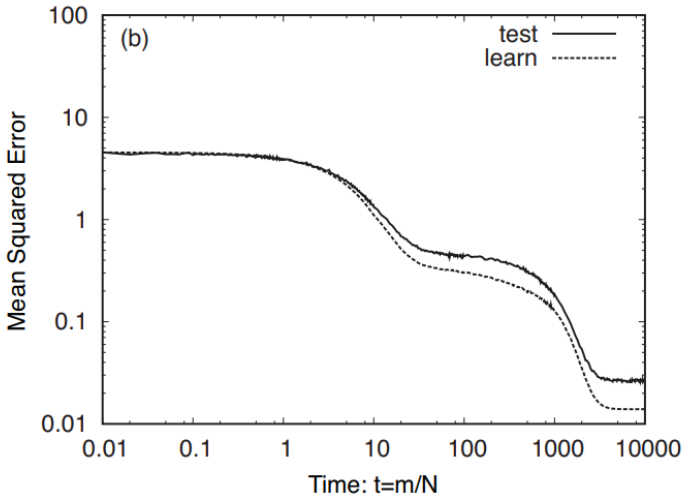
Dropout

- Обучается персептрон со 100 нейронами с Dropout, где $p = 0.5$
- Обучается ансамбль 2 персептронов с 50 нейронами

Ансамбль



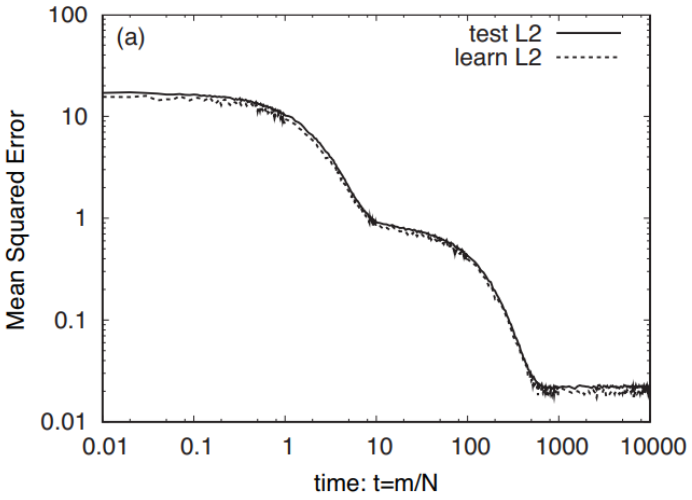
Dropout



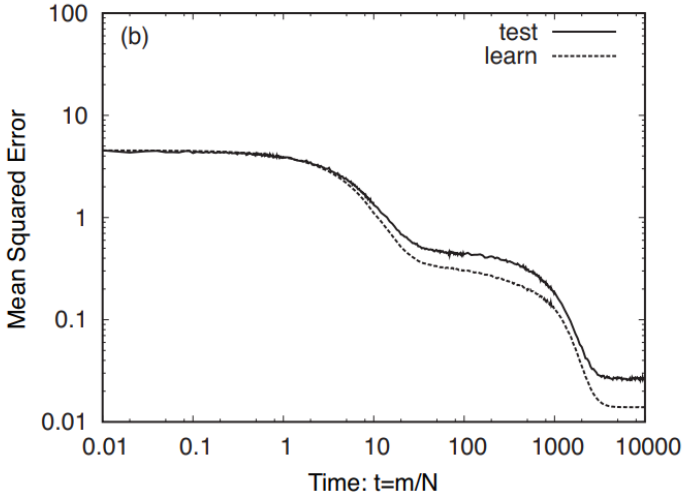
Dropout

- Обучается персептрон со 100 нейронами с Dropout, где $p = 0.5$
- Обучается ансамбль 2 персептронов с 50 нейронами, используя L2 регуляризацию

Ансамбль с L2 регуляризацией



Dropout



Uncertainty Estimation

- Нейронные сети часто выдают слишком уверенные предсказания
- Уверенные неправильные предсказания могут быть небезопасны
- Хотелось бы получить сеть, выдающие откалиброванные предсказания

Uncertainty Estimation

- Надо использовать корректное правило подсчета качества $S(p_\theta, (y, \mathbf{x}))$
- Для него выполняется $S(p_\theta, q) \leq S(q, q)$, где $q(y, \mathbf{x})$ описывает истинное распределение
- Для классификации – все хорошо
- Для регрессии необходимо добавить дополнительный выход – разброс, и надо воспользоваться функцией потерь:

$$-\log p_\theta(y_n|\mathbf{x}_n) = \frac{\log \sigma_\theta^2(\mathbf{x})}{2} + \frac{\left(y - \mu_\theta(\mathbf{x})\right)^2}{2\sigma_\theta^2(\mathbf{x})} + \text{constant}$$

Uncertainty Estimation

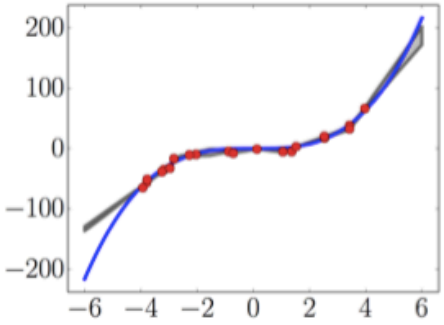
Получаем следующий алгоритм:

1. Обучаем *M* моделей, используя корректную оценку качества
2. Усредняем полученные предсказания

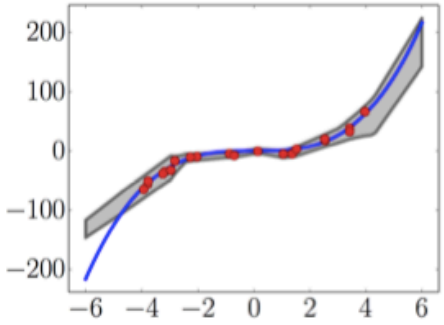
- Для классификации – опять все хорошо
- Для регрессии: $\sigma_*^2(\mathbf{x}) = M^{-1} \sum_m \left(\sigma_{\theta_m}^2(\mathbf{x}) + \mu_{\theta_m}^2(\mathbf{x})\right) - \mu_*^2(\mathbf{x})$

Uncertainty Estimation

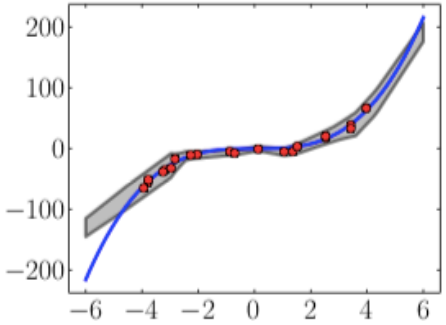
Результаты экспериментов для регрессии



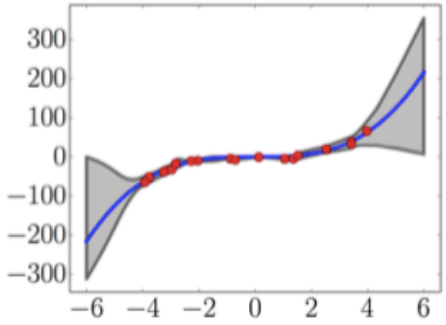
Ансамбль, MSE



Одна модель, NLL



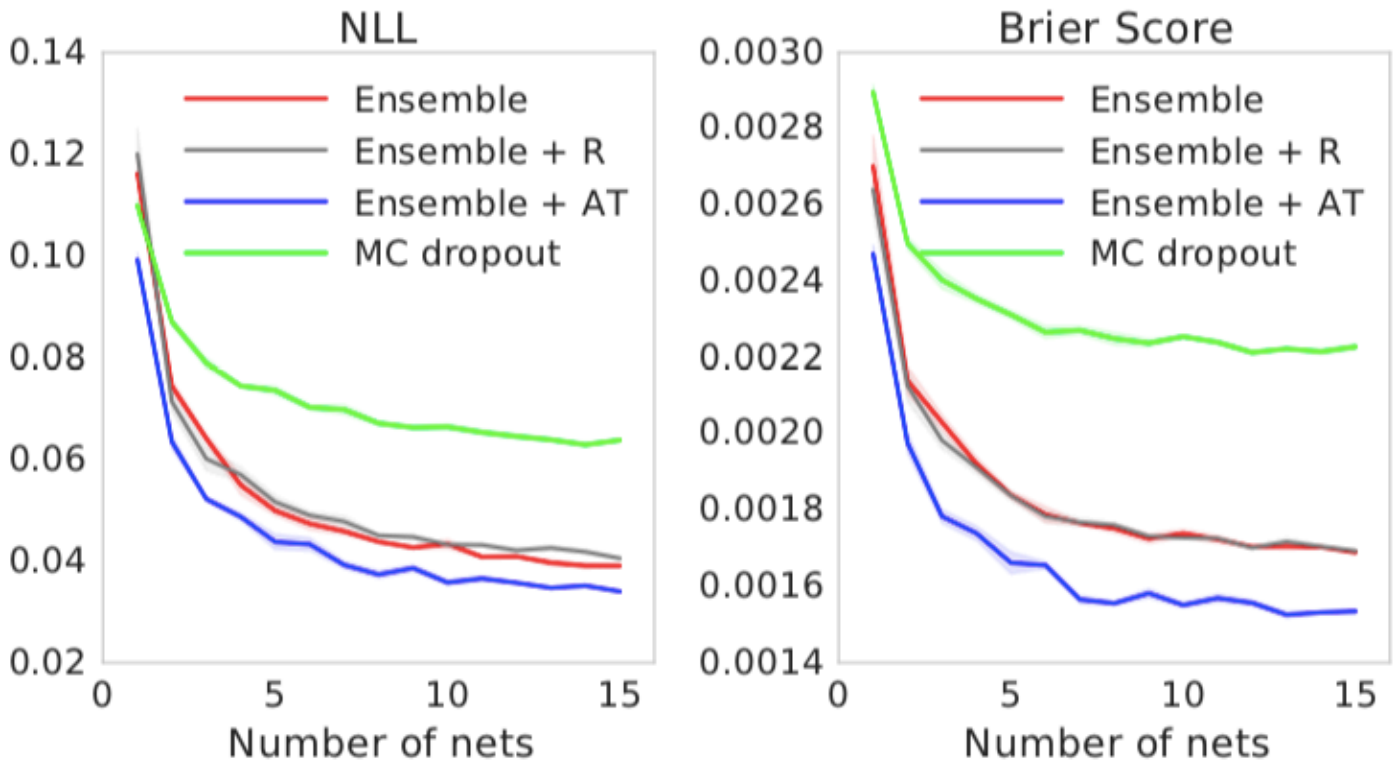
Одна модель + AT, NLL



Ансамбль + AT, NLL

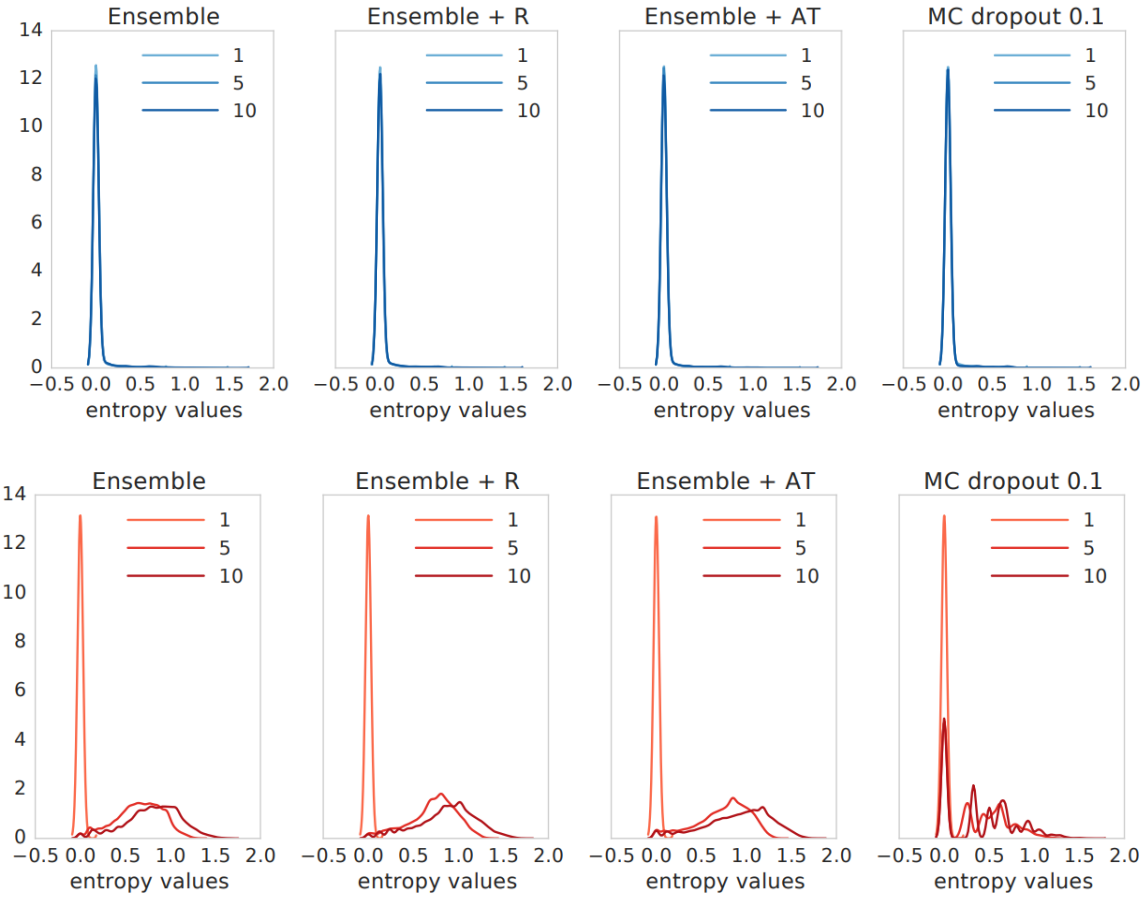
Uncertainty Estimation

Результаты экспериментов для классификации



Uncertainty Estimation

Гистограммы энтропии моделей, обученных на MNIST



Вопросы

1. Зачем использовать ансамбли нейронных сетей? Опишите какой-нибудь способ ансамблирования.
2. Опишите метод построения Snapshot Ensemble. Чем FGE отличается от Snapshot Ensemble?
3. Как нужно обучить ансамбль нейронных сетей в задаче регрессии для Predictive Uncertainty Estimation?

Источники

- Ensemble Learning Methods for Deep Learning Neural Networks
- Horizontal and Vertical Ensemble with Deep Representation for Classification
- Snapshot Ensembles: Train 1, get M for free
- Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs
- Averaging Weights Leads to Wider Optima and Better Generalization
- Analysis of dropout learning regarded as ensemble learning
- Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles