Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs

Лишуди Дмитрий Андреевич, 193

Введение

Рельеф функции потерь нейросетей

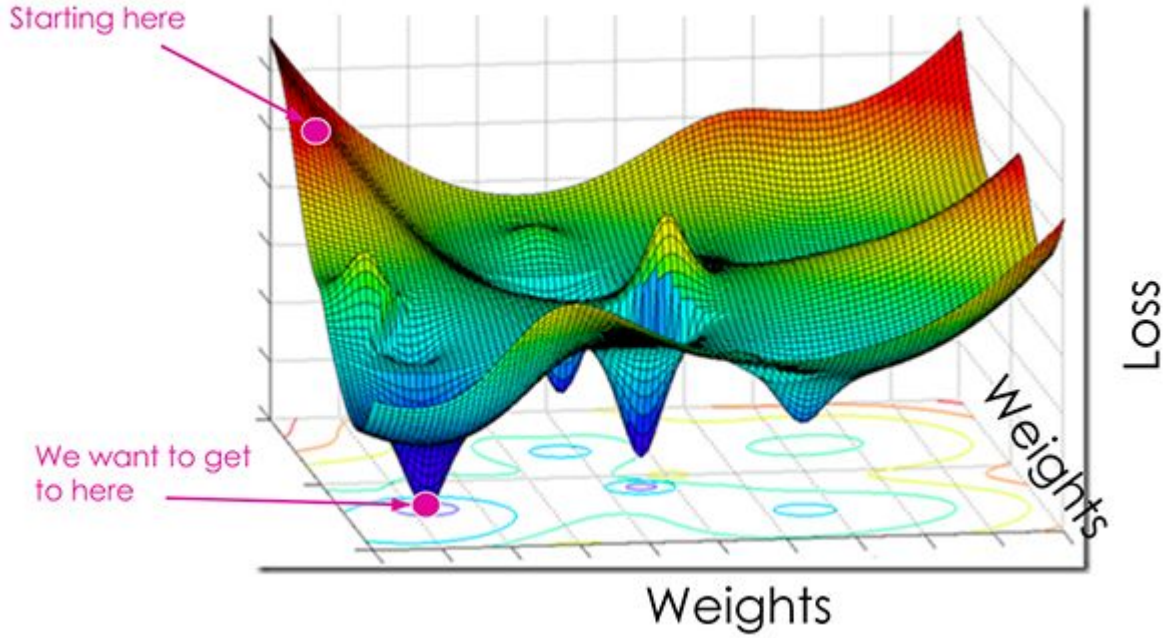

Fig.1: Пример поверхности функции потерь для двух параметров

Вспоминаем МО1

- В нейросетях огромное число параметров.
- Число локальных минимумов возрастает по экспоненте от числа параметров.
- Разные инициализации => разные минимумы.
- Значит у нейросетей высокий разброс (variance) и малое смещение.
- Вспоминаем МО1: в таком случае хорошо работает усреднение моделей.

Ансамбль нейросетей

- Разные нейросети хорошо обрабатывают разные случаи.
- Ансамбль улучшает обобщающую способность.
- Чем слабее модели коррелируют, тем лучше работает ансамблирование.
- Чаще всего вывод ансамбля - среднее или выбор большинством голосов.
- В соревнованиях в kaggle в вершине рейтинга обычно стоят ансамбли.

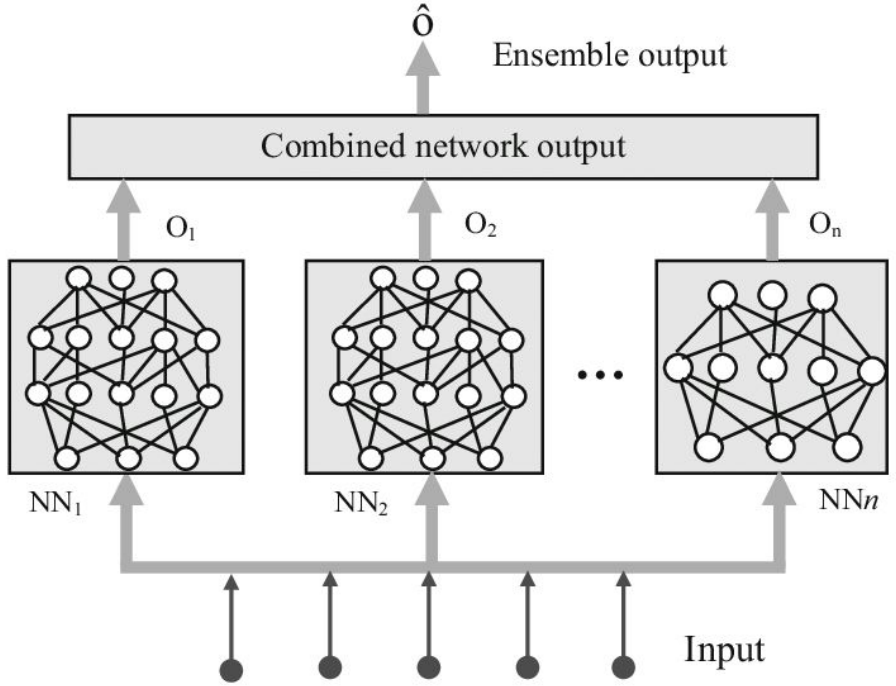

Fig.2: Схема ансамбля нейросетей

Проблемы ансамблей

1) **Разнообразие моделей:** Чтобы ансамбли работали хорошо, модели должны быть разнородными (в идеале - модели не коррелируют). Из-за этого эффективность ансамблей затухает с их размером.
2) **Требовательность к ресурсам:** Чтобы получить ансамбль из N моделей, придётся обучить N нейросетей и потратить в N раз больше ресурсов. При этом эффективность возрастает непропорционально слабее.

Snapshot ансамблирование

Snapshot Ensembles (SSE)
(2017)

Snapshot-ансамбли

- Идея: при обучении одной модели обходим несколько локальных минимумов.
- Попадая в минимум делаем *снимок* (*snapshot*) - сохраняем веса модели.
- В конце строим ансамбль по нескольким лучшим *снимкам*.

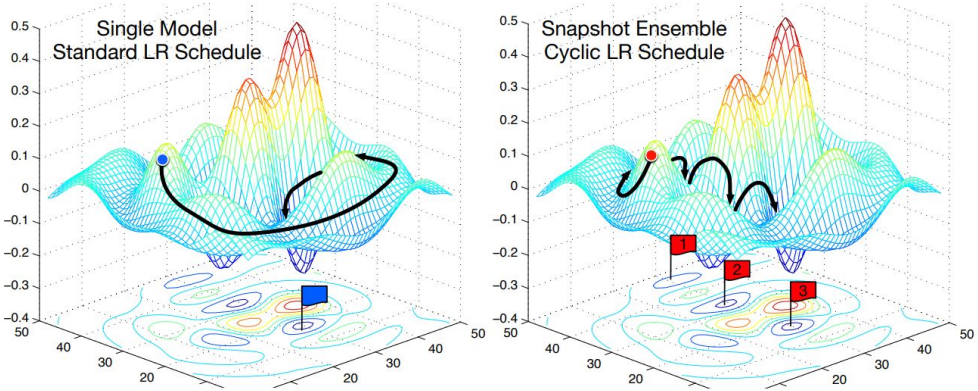

Fig.3:
Слева: траектория SGD при обычном расписании lr.
Справа: траектория Snapshot ансамбля с циклическим расписанием lr

Как этого добиться?

- Обычно нейросеть обучают несколько сотен эпох с расписанием lr.
- Низкая ошибка на тесте достигается после большого падения lr.
- Идея: будем уменьшать lr быстрее, получая немного худший минимум.
- Сохраняем веса, возращаем начальный lr, начинаем заново.
- В конце усредняем softmax слои лучших (последних) моделей.

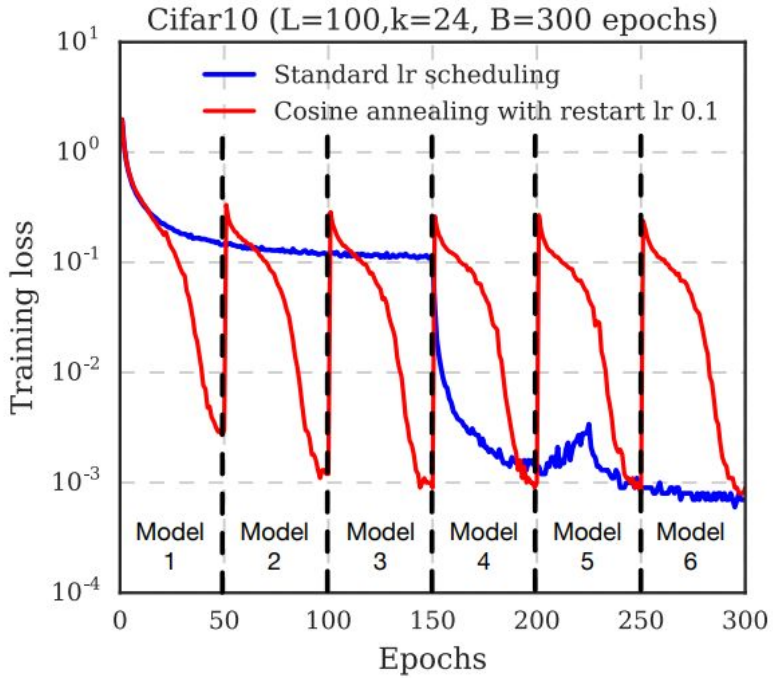

Fig.4: Значения потерь на тесте для обычной модели и Snapshot-модели

Косинусное расписание lr

- Для одной модели:

$$\alpha(t) = \frac{\alpha_0}{2}\left(\cos\left(\pi\frac{t}{T}\right)+1\right)$$

- Для M моделей:

$$\alpha(t) = \frac{\alpha_0}{2}\left(\cos\left(\pi\frac{mod(t,[T/M])}{[T/M]}\right)+1\right)$$

α(t) - lr на итерации (батче) t.
T - суммарное количество итераций.

Результаты Snapshot

Вариации Snapshot Ensemble:

- NoCycle Snapshot - использует расписание lr обычной модели, но равномерно сохраняет веса как SSE.
- SingleCycle Ensembles - после каждого итерации цикла lr реинициализируем веса.

	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**	-	-

Fig.5: Результаты SSE и его вариаций в сравнении с обучением одной модели и Dropout

Быстрое геометрическое ансамблирование

Fast Geometric Ensembling (FGE)
(2018)

Поверхности потерь

- Возможно какие-то хорошие свойства поверхности функции потерь позволят легче получать ансамбли?
- Утверждается, что локальные оптимумы соединяются поверхностью с малыми потерями.

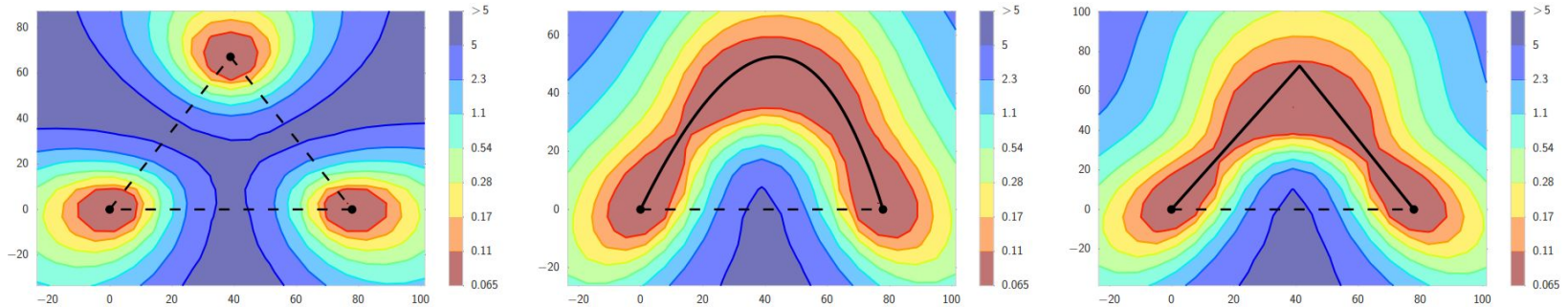

Fig.5: Срезы поверхности кросс-энтропии разными плоскостями в пространстве параметров. Горизонтальная ось совпадает.
Слева: Локальные оптимумы для наборов весов w_1, w_2, w_3.
В центре: Кривая Безье соединяющая w_1 и w_2 в другой плоскости в области малых потерь.
Справа: Равнобедренная ломаная соединяющая w_1 и w_2 в другой плоскости в области малых потерь.

Находим пути между модами

- Значения параметров в этих соединяющих областях можно использовать для построения ансамблей!
- Хотим соединить непрерывной кусочно-гладкой кривой ϕ_θ, θ - параметр.

$$\phi_\theta(0) = w_1; \quad \phi_\theta(1) = w_2$$

- Минимизируем средние потери модели по кривой:

$$\hat{\ell}(\theta) = \frac{\int \mathcal{L}(\phi_\theta) d\phi_\theta}{\int d\phi_\theta} = \frac{\int_0^1 \mathcal{L}(\phi_\theta(t)) \|\phi'_\theta(t)\| dt}{\int_0^1 \|\phi'_\theta(t)\| dt} = \int_0^1 \mathcal{L}(\phi_\theta(t)) q_\theta(t) dt = \mathbb{E}_{t \sim q_\theta(t)} \Big[\mathcal{L}(\phi_\theta(t)) \Big]$$

$$q_\theta(t) = \|\phi'_\theta(t)\| \cdot \left(\int_0^1 \|\phi'_\theta(t)\| dt \right)^{-1}$$

Находим пути между модами

- Градиентный спуск не сработает, ведь распределение зависит от θ.
- Модифируем потери кривой считая, что q_θ – равномерное:

$$\ell(\theta) = \int_0^1 \mathcal{L}(\phi_\theta(t))dt = \mathbb{E}_{t\sim U(0,1)}\mathcal{L}(\phi_\theta(t))$$

- Это те же потери, если ϕ_θ(t) - ломаная из двух отрезков линейных по t.
- Семплируем t равномерно, оптимизируем θ градиентным спуском.

Параметризация кривых

В работе используются два вида кривых:

- Ломаные из двух равных отрезков:

$$\phi_\theta(t) = \begin{cases} 2\left(t\theta + (0.5 - t)\hat{w}_1\right), & 0 \le t \le 0.5 \\ 2\left((t - 0.5)\hat{w}_2 + (1 - t)\theta\right), & 0.5 \le t \le 1. \end{cases}$$

- Квадратичные кривые Безье:

$$\phi_\theta(t) = (1-t)^2\hat{w}_1 + 2t(1-t)\theta + t^2\hat{w}_2, \;\; 0 \le t \le 1.$$

Эксперименты нахождения путей

Модель - ResNet-164. Датасет - CIFAR-100.

Функция потерь - кросс-энтропия с l_2 регуляризацией

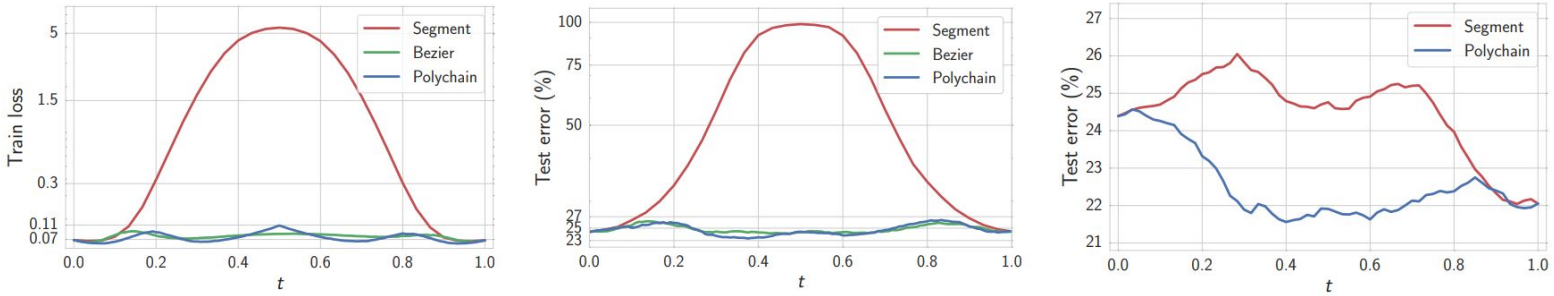

Fig.6: Результаты для отрезка, кривой Безье и ломаной.
Слева: Потери на тренировочной выборке.
В центре: Ошибка на тесте модели с весами ϕ_θ(t).
Справа: Ошибка на тесте на ансамбле моделей с весами w_1 и ϕ_θ(t).

Быстрое геометрическое ансамблирование

- Знаем, что оптимальные веса соединены оптимальными областями.
- Хотелось бы собрать оптимальные веса, но не учить модель много раз.
- Идея: один раз достаточно хорошо обучим модель, а затем постараемся найти параметры из соединяющих областей.

Быстрое геометрическое ансамблирование

1. Обучаем 1 очень хорошую модель (тратим ~80% бюджета на обучение)
2. Находим модели из области малых потерь. Для этого делаем сначала шаги с большим lr, затем - с маленьким по линейному расписанию, всего *c* итераций:

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

$$t(i) = \frac{1}{c}(\mathrm{mod}(i - 1, c) + 1)$$

3. Повторяем шаг 2 нужное количество раз

Визуализация FGE

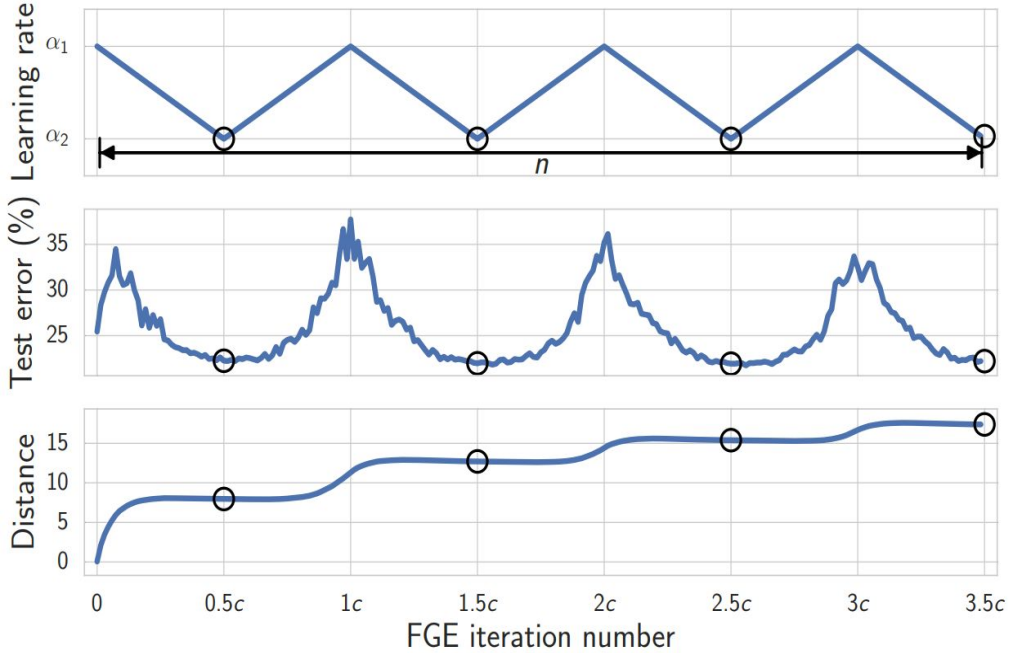

Fig.7: Значения lr (**сверху)**, тестовой ошибки (**в центре)** и l_2 расстояния (**снизу)** в ходе циклических итераций.

Отличия алгоритмов FGE и SSE

Snapshot Ensemble

- M раз повторяем цикл lr, на каждую модель поровну ресурсов.
- Большие шаги: длина цикла 20-40 эпох, lr - косинусный.
- Евклидово расстояние между весами ~40 (ResNet-164; CIFAR-100).

Fast Geometric Ensemble

- Тратим бóльшую часть ресурсов на тренировку одной хорошей модели.
- Затем делаем малые шаги: линейный циклический lr по 2-4 эпохи.
- Евклидово расстояние между весами ~7 (ResNet-164; CIFAR-100).

Результаты FGE

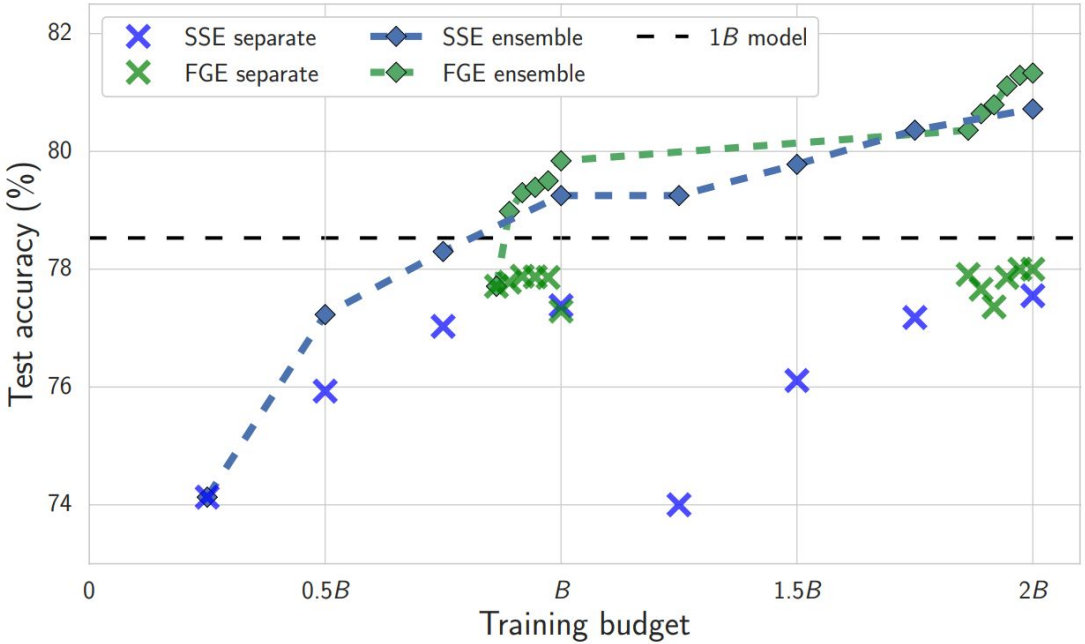

Fig.8: Точность на тестовой выборке для базовой модели, Fast Geometric Ensemble (**FGE**) и Snapshot Ensemble (**SSE**) в зависимости от бюджета вычислений

FGE приближает оптимальные области

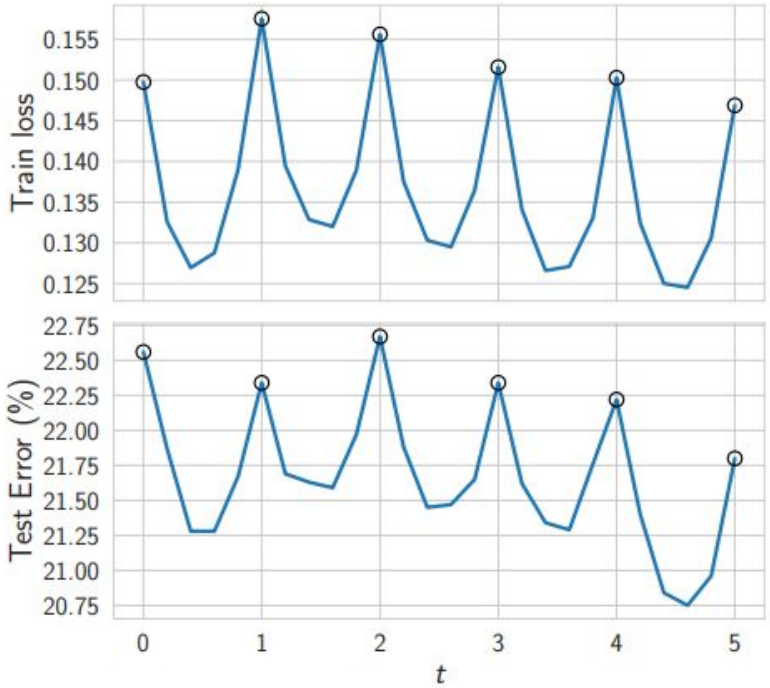

Fig.9: Потери на тренировочной выборке и ошибка на тесте для ломаной, соединяющей веса моделей FGE

Сравнение с моделями в независимых ансамблях

- Из-за близостей моделей страдает разнообразие.

  Для ResNet-164 на CIFAR-100:

  - Две независимо обученные модели: **20%** предсказаний отличаются.
  - Две нейросети из FGE: **15%** предсказаний отличаются.
- Сами по себе модели также чуть менее эффективны:
  - Просто обученная модель: **78.5%** правильных предсказаний
  - Одна модель из FGE: **78.0%** правильных предсказаний
- Тем не менее FGE даёт большой выигрыш в ресурсах!

Сравнение с другими методами ансамблирования

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

Fig.10: Частота ошибок для независимых ансамблей (**Ind**), Snapshot Ensembles (**SSE**) и Fast Geometric Ensembles (**FGE**) в зависимости от бюджета вычислений, модели и датасета.

Спасибо за внимание!