Averaging Weights Leads to Wider Optima and Better Generalization

Шешукова Марина

22 февраля 2022

Полезные заметки

Предположим у нас есть три вектора весов w_1, w_2, w_3. Положим

$$u = (w_2 - w_1),$$

$$v = (w_3 - w_1) - \langle w_3 - \ w_1, w_2 - \ w_1 \rangle / \|w_2 - w_1\| 2 \cdot (w_2 - w_1).$$

Тогда векторы $\hat{u} = u/\|u\|$, $\hat{v} = v/\|v\|$ образуют ортонормированный базис плоскости, которая содержит w_1, w_2, w_3.
Для визуализации потерь на этой плоскости, определим декартову систему координат с базисом $\hat{u}, \hat{v}$ и оценим сети соответсвующие каждой точке на плоскости. Точка P с координатами (x, y) на плоскости задана, как $P = w_1 + x \cdot \hat{u} + y \cdot \hat{v}$.

Анализ траекторий SGD

Для того чтобы исследовать область весового пространства используют расписание длины шага. Для циклического шага в статье используют следующую формулу: на итерации i

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

Базовые $\alpha_1 \geqslant \alpha_2$ и длина цикла c являются гиперпараметрами. Если взять $\alpha_1 = \alpha_2$, то получиться константная длина шага.

Анализ траекторий SGD

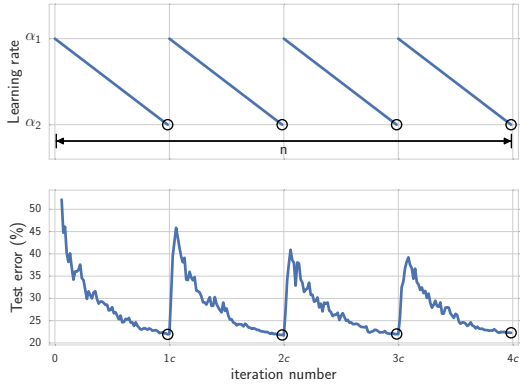


Рис.: **Верхняя картинка**: циклическое расписание длины шага. **Нижняя картинка**: ошибка на тесте при циклическом расписании длины шага при использовании модели Preactivation-ResNet-164 на датасете CIFAR-100.

Анализ траекторий SGD

- Запускаем SGD с циклической и константной длиной шага для предобученной точки весов на модели Preactivation ResNet-164 и датасете CIFAR-100.
- Строим плоскость по первой, последней и средней точке весов для каждой траектории в пространстве весов.
- Строим ошибку на обучении и тесте для каждой из этих плоскостей.
- Проецируем оставшиеся точки траекторий на полученные плоскости.

Анализ траекторий SGD

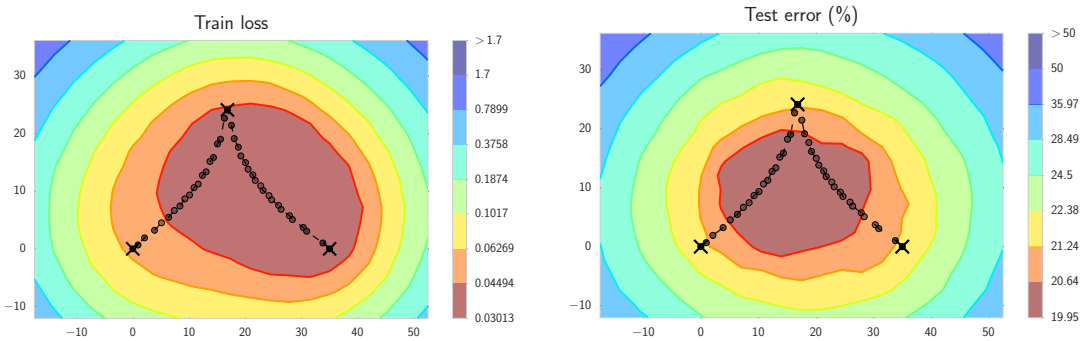


Рис.: Логистическая функция потерь с $L-2$ регуляризацией и циклической длиной шага.

Анализ траекторий SGD

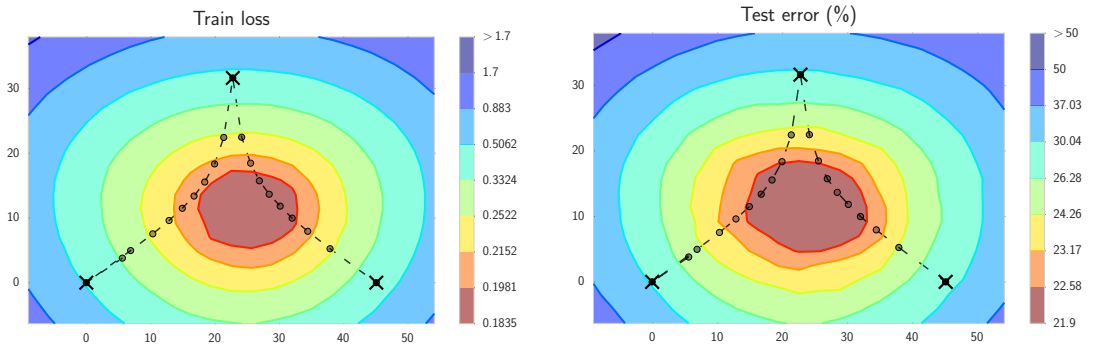


Рис.: Логистическая функция потерь с $L-2$ регуляризацией и константной длиной шага.

Анализ траекторий SGD

Выводы:

- Точки траектории (кроме первой, последней и средней) не лежат в построенной плоскости, поэтому для них невозможно определить ошибку на обучении и тесте.
- Оба метода исследуют точки, близкие к периферии набора высокопроизводительных сетей.
- Поверхности потерь на обучении и тесте похожи, но не совпадают в точности. Этот сдвиг между обучением и тестом предполагает, что более центральные точки могут привести к лучшему обобщению.

Алгоритм SWA

Algorithm 1 Stochastic Weight Averaging

Require:
weights $\hat{w}$, LR bounds α_1, α_2,
cycle length c (for constant learning rate $c = 1$), number of iterations n

Ensure: w_{SWA}

$w \leftarrow \hat{w}$ {Initialize weights with $\hat{w}$}
$w_{\text{SWA}} \leftarrow w$
for $i \leftarrow 1, 2, \ldots, n$ **do**
  $\alpha \leftarrow \alpha(i)$ {Calculate LR for the iteration}
  $w \leftarrow w - \alpha \nabla \mathcal{L}_i(w)$ {Stochastic gradient update}
  if $\text{mod}(i, c) = 0$ **then**
    $n_{\text{models}} \leftarrow i/c$ {Number of models}
    $w_{\text{SWA}} \leftarrow \frac{w_{\text{SWA}} \cdot n_{\text{models}} + w}{n_{\text{models}} + 1}$ {Update average}
  end if
end for
{Compute BatchNorm statistics for w_{SWA} weights}

Рис.: Алгоритм SWA

Пакетная нормализация

Если DNN использует пакетную нормализацию, то мы выполняем один дополнительный проход по данным, чтобы вычислить скользящее среднее и стандартное отклонение активаций для каждого слоя сети с весами w_{SWA} после завершения обучения, поскольку эти статистические данные не собираются во время обучения. Для большинства библиотек глубокого обучения, таких как PyTorch или Tensorflow, обычно можно собирать эту статистику, выполняя прямой проход по данным в режиме обучения.

Время и память.

- Во время обучения нужно хранить копию скользящего среднего веса DNN
- В потреблении памяти при хранении DNN преобладают ее активации, а не ее веса, и поэтому процедура SWA лишь немного увеличивает ее, даже для больших DNN (например, порядка 10 процентов)
- Во время обучения дополнительно тратится время только на обновление среднего веса
- Мы применяем эту операцию не чаще одного раза за эпоху, таким образом, SWA и SGD требуют практически одинакового объема вычислений.

Оптимумы SWA и SGD

Идея: поверхности потерь при обучении и тесте сдвинуты, поэтому желательно сходиться «широким» оптимумам, то есть к тем, которые остаются приблизительно оптимальными при небольших возмущениях.

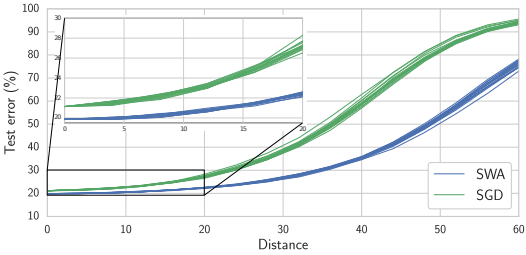
Оптимумы SWA и SGD

- Пусть w_{SWA} и w_{SGD} обозначают веса DNN, полученные с помощью SWA и SGD соответственно.
- Рассмотрим лучи

$$w_{\text{SWA}}(t, d) = w_{\text{SWA}} + t \cdot d,$$
$$w_{\text{SGD}}(t, d) = w_{\text{SGD}} + t \cdot d,$$

- Нарисуем ошибку на обучении и тесте для 10-ти различных направлений d_i.

Оптимумы SWA и SGD



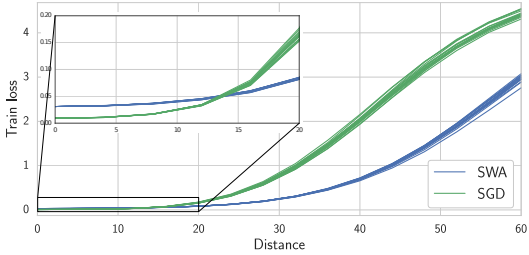


Рис.: Ошибка на тесте и обучении для рандомных лучей. Используется логистическая функция потерь с $L-2$ регуляризацией на модели Preactivation ResNet-164 и датасете CIFAR-100.

Оптимумы SWA и SGD

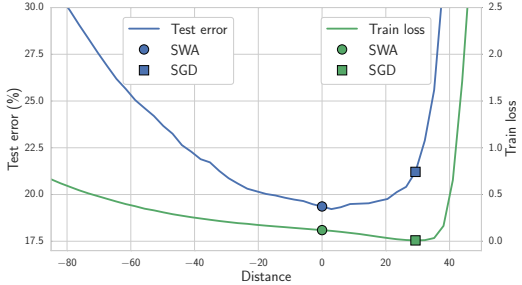
Выводы:

- На обучающей выборке SGD имеет меньшие потери, а вот на тесте SWA имеет меньшие потери (при $t = 0$).
- От w_{SWA} нужно отступить значительно больше, чтобы получить такое же изменение ошибки, как на w_{SGD}.
- График w_{SGD} имеет точку перегиба, которая отсутствует у графика w_{SWA}

Оптимумы SWA и SGD

Теперь рассмотрим отрезок соединяющий w_{SWA} и w_{SGD}:

$$w(t) = t \cdot w_{\text{SGD}} + (1 - t) \cdot w_{\text{SWA}} .$$



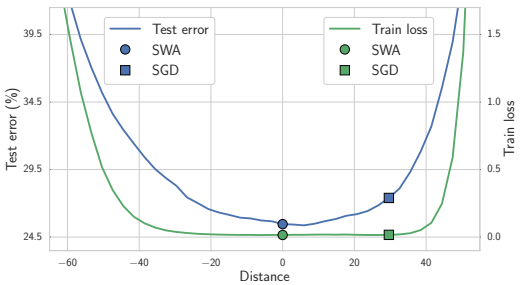


Рис.: Ошибка на тесте и обучении на прямой, проходящей через w_{SWA} и w_{SGD}. Используется логистическая функция потерь с $L - 2$ регуляризацией. **Левая**: Preactivation ResNet-164, CIFAR-100. **Правая**: VGG-16, CIFAR-100.

Оптимумы SWA и SGD

Выводы:

- Графики на тестовой и обучающей выборках действительно смешены относительно друг друга, поэтому оптимальная точка w_{SGD} на обучающей выборке далека от оптимальной на тестовой выборке.
- w_{SWA} находится в области «широкого» оптимума
- w_{SGD} может находиться в области оптимума с крутым подъемом в каком-то из направлений, в результате чего ошибка на тесте может быть хуже.

Fast Geometric Ensembling (FGE)

Algorithm 1 Fast Geometric Ensembling

Require:
weights $\hat{w}$, LR bounds α_1, α_2,
cycle length c (even), number of iterations n
Ensure: ensemble
$w \leftarrow \hat{w}$ {Initialize weight with $\hat{w}$}
ensemble $\leftarrow [\,]$
for $i \leftarrow 1, 2, \ldots, n$ **do**
$\alpha \leftarrow \alpha(i)$ {Calculate LR for the iteration}
$w \leftarrow w - \alpha \nabla \mathcal{L}_i(w)$ {Stochastic gradient update}
if $\mathrm{mod}(i, c) = c/2$ **then**
ensemble $\leftarrow$ ensemble $+ [w]$ {Collect weights}
end if
end for

Рис.: Алгоритм FGE

Обозначения

- Пусть $f(\cdot)$ обозначает предсказания нейронной сети, параметризованной весами w.
- Будем считать, что $f(\cdot)$ выдает скаляр и дважды непрерывно дифференцируема по w.
- Пусть w_i — точки предложенные *FGE*, они сосредоточены вокруг своего среднего $w_{\text{SWA}} = \frac{1}{n}\sum_{i=1}^{n} w_i$.
- Также обозначим за $\Delta_i = w_i - w_{\text{SWA}}$.
- Заметим, что $\sum_{i=1}^{n} \Delta_i = 0$.
- FGE выдает

$$\bar{f} = \frac{1}{n}\sum_{i=1} nf(w_i).$$

Связь SWA с FGE

- Запишем ряд Тейлора:

$$f(w_j) = f(w_{\text{SWA}}) + \langle \nabla f(w_{\text{SWA}}), \Delta_j \rangle + O(\|\Delta_j\|2),$$

- Разница между усреднением весов и усреднением предсказания:

$$\begin{aligned} \bar{f} - f(w_{\text{SWA}}) &= \frac{1}{n} \sum_{i=1}^{n} \left(\langle \nabla f(w_{\text{SWA}}), \Delta_i \rangle + O(\|\Delta_i\|2) \right) \\ &= \left\langle \nabla f(w_{\text{SWA}}), \frac{1}{n} \sum_{i=1}^{n} \Delta_i \right\rangle + O(\Delta 2) = O(\Delta 2), \end{aligned}$$

где $\Delta = \max_{i=1}^{n} \|\Delta_i\|$.

Эксперименты SGD, FGE, SWA

Table 1: Accuracies (%) of SWA, SGD and FGE methods on CIFAR-100 and CIFAR-10 datasets for different training budgets. Accuracies for the FGE ensemble are from Garipov et al. [2018].

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

Рис.: Эксперименты

Расписание шага и SWA

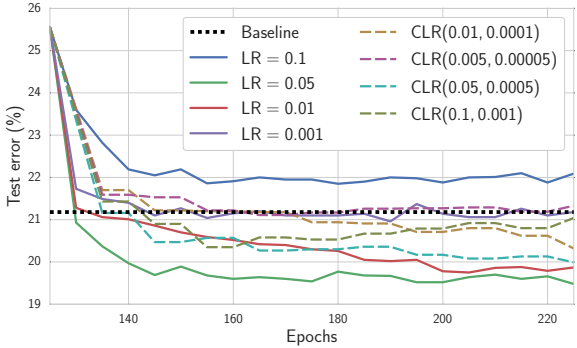


Рис.: Ошибка на тесте SWA с различным расписанием длины шага с моделью Preactivation ResNet-164 на датасете CIFAR-100.

Список литературы.

- https://arxiv.org/pdf/1802.10026.pdf
- https://arxiv.org/pdf/1803.05407.pdf