ТЕХНОСФЕРА

Лекция 5
Улучшение сходимости нейросетей

Байгушев Данила

23 сентября 2019 г.

Улучшение сходимости

Ускорение сходимости

- Инициализация (Xavier, He)
- Нормализация (Batch Normalization, Layer normalization)

Борьба с переобучением

- Регуляризация (Dropout, DropConnect)

Инициализация весов

Xavier (Glorot)

Рассмотрим нечетную функцию с единичной производной в нуле в качестве активации (напр. $\tanh$)

- Хотим начать из линейного региона, чтобы избежать затухающих градиентов

$$z^{i+1} = f(\underbrace{z^i W^i}_{s^i})$$

$$\mathbb{D}[z^i] = \mathbb{D}[x] \prod_{k=0}^{i-1} n_k \mathbb{D}[W^k]$$

$$\mathbb{D}[\frac{\partial L}{\partial s^i}] = \mathbb{D}[\frac{\partial L}{\partial s^d}] \prod_{k=i}^{d} n_{k+1} \mathbb{D}[W^k]$$

Где n_i — размерность i-того слоя

Xavier (Glorot)

Хорошая инициализация:

$$\forall (i,j) \begin{cases} \mathbb{D}[z^i] = \mathbb{D}[z^j] \\ \mathbb{D}[\frac{\partial L}{\partial s^i}] = \mathbb{D}[\frac{\partial L}{\partial s^j}] \end{cases}$$

Это эквивалентно следующему:

$$\forall i \begin{cases} n_i \mathbb{D}[W^i] = 1 \\ n_{i+1} \mathbb{D}[W^i] = 1 \end{cases}$$

Компромисс: $\mathbb{D}[W^i] = \frac{2}{n_i + n_{i+1}}$

$$\boxed{W^i \sim U[-\frac{\sqrt{6}}{\sqrt{n_i + n_{i+1}}}, \frac{\sqrt{6}}{\sqrt{n_i + n_{i+1}}}]}$$

$\mathbb{D}[U(a,b)] = \frac{1}{12}(b-a)^2$

He [1]

Рассмотрим ReLU в качестве активации:

- Функция не симметрична
- Не дифференцируема в нуле

$$\mathbb{D}[z^i] = \mathbb{D}[x](\prod_{k=0}^{i-1} \frac{1}{2} n_k \mathbb{D}[W^k]) \Rightarrow \mathbb{D}[W^k] = \frac{2}{n_k}$$

$$\mathbb{D}[\frac{\partial L}{\partial s^i}] = \mathbb{D}[\frac{\partial L}{\partial s^d}](\prod_{k=i}^{d} \frac{1}{2} n_{k+1} \mathbb{D}[W^k]) \Rightarrow \mathbb{D}[W^k] = \frac{2}{n_{k+1}}$$

Достаточно использовать только первое уравнение:

$$\mathbb{D}[\frac{\partial L}{\partial s^i}] = \mathbb{D}[\frac{\partial L}{\partial s^d}] \prod_{k=1}^{d} \frac{1}{2} n_{k+1} \mathbb{D}[W^k] = \frac{n_2}{n_d} \mathbb{D}[\frac{\partial L}{\partial s^d}]$$

n_2/n_d небольшое для сверточных сетей

$$W^i \sim N(0, \tfrac{2}{n_i})$$

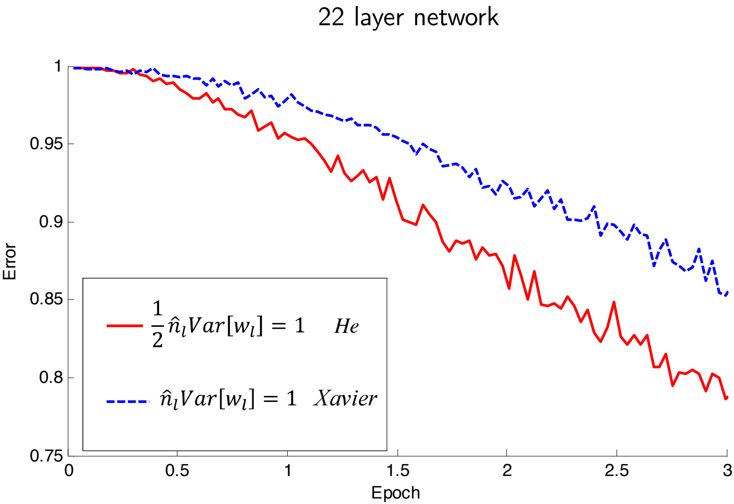
or

$$W^i \sim N(0, \tfrac{2}{n_{i+1}})$$

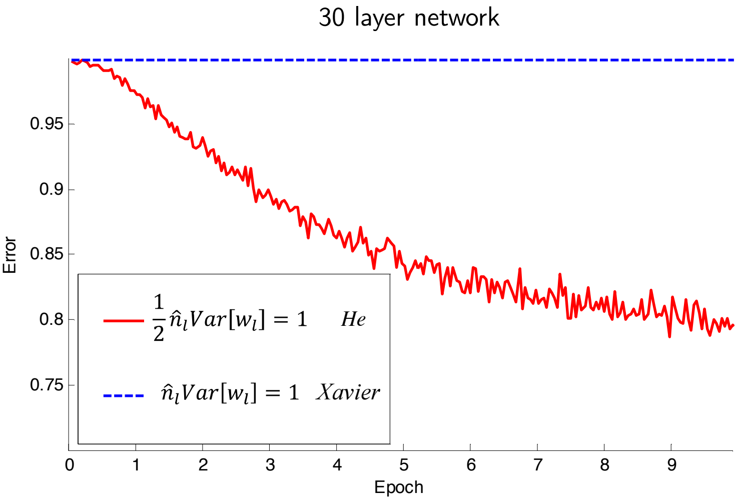
---

[1] `https://arxiv.org/abs/1502.01852`

Xavier против He для ReLU



Xavier против He для ReLU



Ортогональная инициализация[2]

Выберем ортогональную матрицу весов W: $WW^T = I$. Тогда:

- $\|W_i x\| = \|x\|$ — норма сохраняется
- $\langle W_i, W_j \rangle = \delta_{ij}$ — все нейроны делают «разные» преобразования

Что делать для сверточных слоев?

---

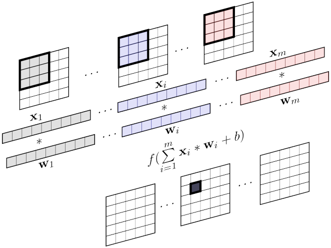
[2] `https://hjweide.github.io/orthogonal-initialization-in-convolutional-layers`

Ортогональная инициализация[2]

Выберем ортогональную матрицу весов W: $WW^T = I$. Тогда:

- $\|W_i x\| = \|x\|$ — норма сохраняется
- $\langle W_i, W_j \rangle = \delta_{ij}$ — все нейроны делают «разные» преобразования

Что делать для сверточных слоев?



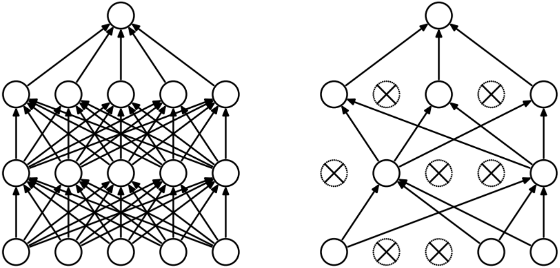
1. Генерируем ортогональную матрицу $W \in \mathbb{R}^{c' \times k^2 c}$
2. Reshape: $K \in \mathbb{R}^{c' \times c \times k \times k}$

---

[2] `https://hjweide.github.io/orthogonal-initialization-in-convolutional-layers`

Регуляризация

Dropout[3]



- С вероятностью p занулим выход нейрона (например, $p = 0.5$)
- В test-time домножаем веса на вероятность сохранения
- Не стоит выкидывать нейроны последнего слоя

---

[3] Dropout: A Simple Way to Prevent Neural Networks from Overfitting N. Srivastava, G. Hinton

Dropout, мотивация

- Борьба с соадаптацией – нейроны больше не могут рассчитывать на наличие соседей
- Биология: не все гены родителей будут присутсвовать у потомков
- Усреднение большого (2^n) числа моделей

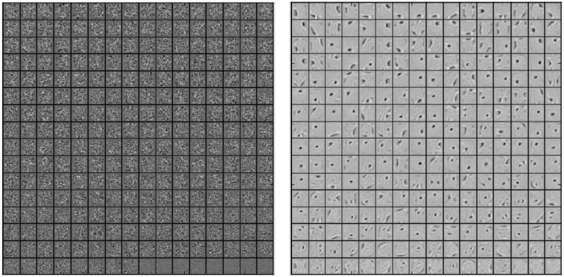
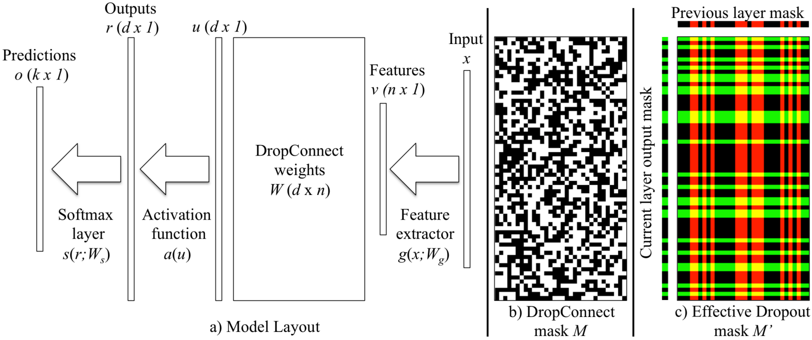


Figure: Выученные признаки на MNIST (автокодировщик с одним скрытым слоем и ReLU в качестве активации). Слева: без Dropout, справа – с Dropout

Dropconnect[4]



a) Model Layout

b) DropConnect mask M

c) Effective Dropout mask M'

- Зануляем не выходы нейронов, а каждый вес по отдельности

---

[4] `https://cs.nyu.edu/~wanli/dropc/dropc.pdf`

Нормализация

Мотивация

- Обычно наблюдается более быстрая сходимость при декорелированных входах
- Whitening: $\hat{\mathbf{x}} = Cov[\mathbf{x}]^{-1/2}(\mathbf{x} - E[\mathbf{x}])$
- Нормализация: $\hat{x}^{(k)} = \frac{x^{(k)} - E[x^{(k)}]}{\sqrt{Var[x^{(k)}]}}$ для каждой размерности

Батч-нормализация [5]

- Covariate shift: изменение распределения входов во время обучения
- Цель — уменьшить covariate shift скрытых слоев
- Нормализуем входы в каждый слой $\hat{x}^{(k)} = \frac{x^{(k)} - \mathbb{E}[x^{(k)}]}{\sqrt{\mathbb{D}[x^{(k)}]}}$
- Статистики $\mathbb{E}x$ и $\mathbb{D}x$ оценим для каждого мини-батча

? Почему этот метод плох для сетей с сигмоидами?

---

[5] `https://arxiv.org/abs/1502.03167`

Батч-нормализация [5]

- Covariate shift: изменение распределения входов во время обучения
- Цель — уменьшить covariate shift скрытых слоев
- Нормализуем входы в каждый слой $\hat{x}^{(k)} = \frac{x^{(k)} - \mathbb{E}[x^{(k)}]}{\sqrt{\mathbb{D}[x^{(k)}]}}$
- Статистики $\mathbb{E}x$ и $\mathbb{D}x$ оценим для каждого мини-батча

? Почему этот метод плох для сетей с сигмоидами?

- Сигмоиды становятся почти линейными ⇒ линейная модель :(
- Доп. параметры: $y^{(k)} = \gamma^{(k)}\hat{x}^{(k)} + \beta^{(k)}$

---

[5] `https://arxiv.org/abs/1502.03167`

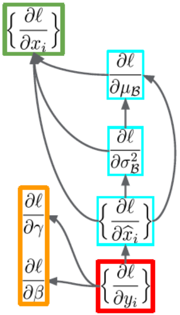
Алгоритм

```
Входы:      Значения $x$ в мини-батче $\mathcal{B} = \{x_i\}_{i=1}^{m}$;
Параметры:  $\gamma$, $\beta$
Выход:      $\{y_i = \mathsf{BN}_{\gamma,\beta}(x_i)\}$

$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^{m} x_i$                                   // среднее мини-батча
$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m-1} \sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2$    // дисперсия мини-батча
$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$   // нормализация
$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathsf{BN}_{\gamma,\beta}(x_i)$            // растяжение и сдвиг
```

Градиент

Можно вычислить градиент при помощи chain rule
Важно помнить, что $\mu_{\mathcal{B}}$ и $\sigma_{\mathcal{B}}^2$ не являются константами



$$\frac{\partial \ell}{\partial \widehat{x}_i} = \frac{\partial \ell}{\partial y_i} \cdot \gamma$$

$$\frac{\partial \ell}{\partial \sigma_{\mathcal{B}}^2} = \sum_{i=1}^m \frac{\partial \ell}{\partial \widehat{x}_i} \cdot (x_i - \mu_{\mathcal{B}}) \cdot \frac{-1}{2}(\sigma_{\mathcal{B}}^2 + \epsilon)^{-3/2}$$

$$\frac{\partial \ell}{\partial \mu_{\mathcal{B}}} = \left(\sum_{i=1}^m \frac{\partial \ell}{\partial \widehat{x}_i} \cdot \frac{-1}{\sqrt{\sigma_{\mathcal{B}}^2+\epsilon}} \right) + \frac{\partial \ell}{\partial \sigma_{\mathcal{B}}^2} \cdot \frac{\sum_{i=1}^m -2(x_i-\mu_{\mathcal{B}})}{m-1}$$

$$\frac{\partial \ell}{\partial x_i} = \frac{\partial \ell}{\partial \widehat{x}_i} \cdot \frac{1}{\sqrt{\sigma_{\mathcal{B}}^2+\epsilon}} + \frac{\partial \ell}{\partial \sigma_{\mathcal{B}}^2} \cdot \frac{2(x_i-\mu_{\mathcal{B}})}{m-1} + \frac{\partial \ell}{\partial \mu_{\mathcal{B}}} \cdot \frac{1}{m}$$

$$\frac{\partial \ell}{\partial \gamma} = \sum_{i=1}^m \frac{\partial \ell}{\partial y_i} \cdot \widehat{x}_i$$

$$\frac{\partial \ell}{\partial \beta} = \sum_{i=1}^m \frac{\partial \ell}{\partial y_i}$$

Предсказание

Во время предсказания батч-нормализация является линейным слоем:

$$\hat{x} = \frac{x - \mathbb{E}[x]}{\sqrt{\mathbb{D}[x] + \epsilon}}$$

$$y = \gamma \cdot \hat{x} + \beta$$

$$y = \frac{\gamma}{\sqrt{\mathbb{D}[x] + \epsilon}} \cdot x + (\beta - \frac{\gamma \mathbb{E}[x]}{\sqrt{\mathbb{D}[x] + \epsilon}})$$

$\mathbb{E}[x]$ и $\mathbb{D}[x]$ вычисляются по всему обучающему множеству.
На практике статистики вычисляются во время обучения экспоненциальным средним: $E_{i+1} = (1 - \alpha) E_i + \alpha E_{\mathcal{B}}$

Batchnorm как регуляризация

- $\frac{\partial BN((aW)u)}{\partial u} = \frac{\partial BN(Wu)}{\partial u}$
- $\frac{\partial BN((aW)u)}{\partial aW} = \frac{1}{a}\frac{\partial BN(Wu)}{\partial W}$

При увеличении весов в a раз, градиент выхода слоя по входу не меняется, а градиент по весам уменьшается в a раз.

Tips

Стоит помнить, что с батч-нормализацией:

- Надо убрать смещения
- Другое расписание learning rate: большее значение в начале обучения и быстрое уменьшение в процессе обучения
- Уменьшить силу Dropout и L_2 регуляризации
- Перемешивать обучающую выборку

Для изображений: нормализация каждого канала (одинаковые среднее и дисперсия вдоль пространственных размерностей)

Обучение

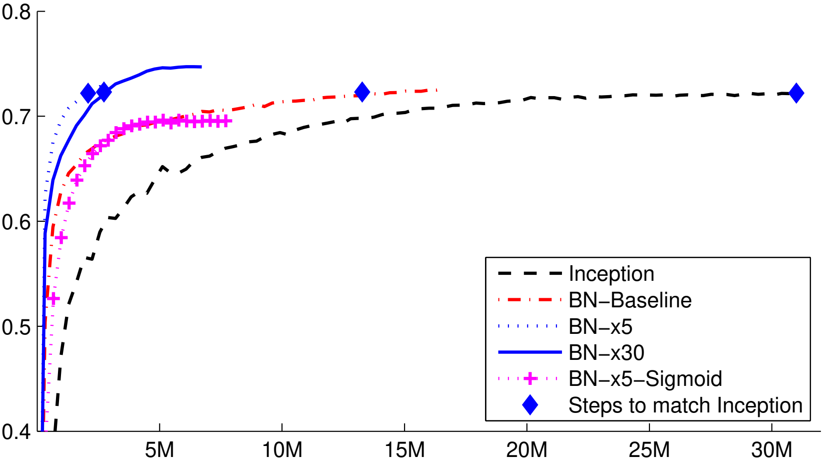
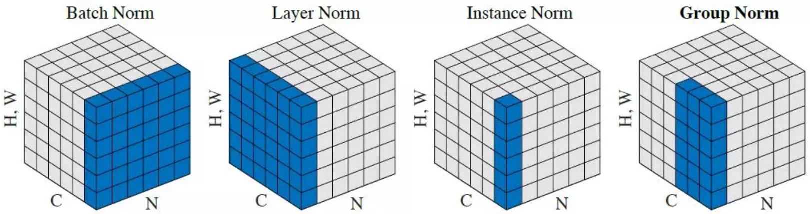


Figure: Обучение Inception с и без батч-нормализации.[6]

---

[6] x30 — увеличение темпа обучения в 30 раз

Layer Normalization [7]



	Weight matrix re-scaling	Weight matrix re-centering	Weight vector re-scaling	Dataset re-scaling	Dataset re-centering	Single training case re-scaling
Batch norm	Invariant	No	Invariant	Invariant	Invariant	No
Weight norm	Invariant	No	Invariant	No	No	No
Layer norm	Invariant	Invariant	No	Invariant	No	Invariant

---

[7] `https://arxiv.org/abs/1607.06450`

Вопросы