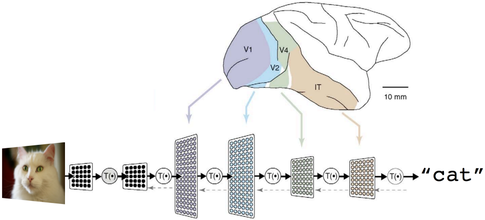
ТЕХНОСФЕРА

Лекция 5
Глубинные нейронные сети, часть 1

Полыковский Даниил

6 марта 2017 г.

Зачем нам глубинные сети?



- Хорошая биологическая мотивация[1]: каждый следующий слой выучивает новый уровень абстракции данных [2] (например штрихи, пятна, поверхности, объекты);
- Хорошие торетические свойства[3].

[1] Hubel and Wiesel 1962; Serre et al. 2005; Ranzato et al. 2007
[2] Palmer 1999; Kandel et al. 2000
[3] Learning multiple layers of representation (G. Hinton, 2007)

Паралич сети, эксперимент

input [841]	layer -5	layer -4	layer -3	layer -2	layer -1	output
neurons	100	100	100	100	100	26
|grad|	6.2e-8	2.2e-6	1.6e-5	1.1e-4	7e-4	0.015

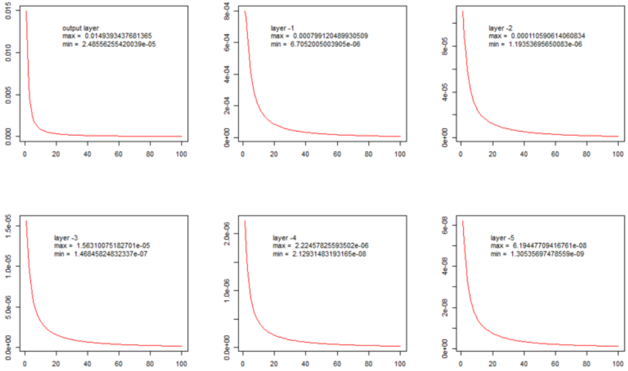
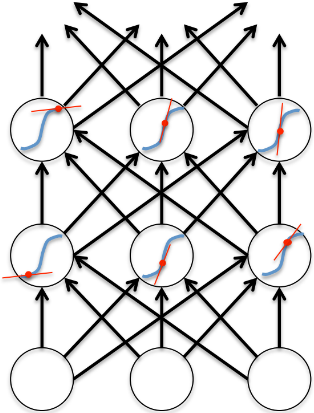


Рис.: Средний модуль градиента в различных слоях

Backprop, прямой проход



Прямой проход в нейросети[4]

- $y^{(k)} = \sigma(x^{(k)}(W^{(k)})^T)$
- выходные значения каждого нейрона лежат в интервале $(0, 1)$

[4] https://class.coursera.org/neuralnets-2012-001/lecture

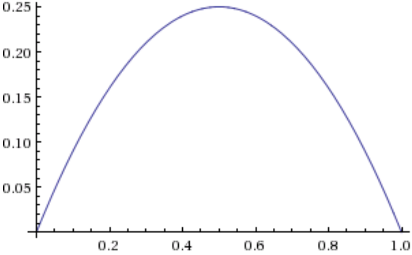
Backprop, затухание градиента

Рассмотрим в качестве функции активации логистическую функцию:

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

$$\frac{\partial \sigma(z)}{\partial z} = \sigma(z)\cdot(1-\sigma(z))$$

Построим график значений производной:



- максимум равен $\sigma_{\max} = \frac{1}{4}$

Backprop, затухание градиента

Рассмотрим простую сеть (один нейрон в каждом слое):

Прямой проход:

$x = z_0$

$z_k = \sigma(z_{k-1} w_k)$

$y = z_5$

Вычислим градиенты весов для $L(y, t) = \frac{1}{2}(y_j - t_j)^2$:

$$\frac{\partial E}{\partial z_4} \quad =$$

Backprop, затухание градиента

Рассмотрим простую сеть (один нейрон в каждом слое):

Прямой проход:

$x = z_0$

$z_k = \sigma(z_{k-1} w_k)$

$y = z_5$

Вычислим градиенты весов для $L(y,t) = \frac{1}{2}(y_j - t_j)^2$:

$$\frac{\partial E}{\partial z_4} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial z_4} = \overbrace{(y-t)}^{\leq 2}\overbrace{\sigma'(w_5 z_4)}^{\leq \frac{1}{4}} w_5 \leq 2 \cdot \frac{1}{4} w_5$$

$$\frac{\partial E}{\partial z_3} =$$

Backprop, затухание градиента

Рассмотрим простую сеть (один нейрон в каждом слое):

Прямой проход:

$x = z_0$

$z_k = \sigma(z_{k-1} w_k)$

$y = z_5$

Вычислим градиенты весов для $L(y,t) = \frac{1}{2}(y_j - t_j)^2$:

$$\frac{\partial E}{\partial z_4} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial z_4} = \overbrace{(y-t)}^{\leq 2}\overbrace{\sigma'(w_5 z_4)}^{\leq \frac{1}{4}} w_5 \leq 2 \cdot \frac{1}{4} w_5$$

$$\frac{\partial E}{\partial z_3} = \frac{\partial L}{\partial z_4}\frac{\partial z_4}{\partial z_3} \leq 2 \cdot (\frac{1}{4})^2 w_4 w_5$$

$$\frac{\partial E}{\partial x} =$$

Backprop, затухание градиента

Рассмотрим простую сеть (один нейрон в каждом слое):

Прямой проход:

$x = z_0$

$z_k = \sigma(z_{k-1} w_k)$

$y = z_5$

Вычислим градиенты весов для $L(y, t) = \frac{1}{2}(y_j - t_j)^2$:

$$\frac{\partial E}{\partial z_4} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial z_4} = \overbrace{(y - t)}^{\leq 2}\overbrace{\sigma'(w_5 z_4)}^{\leq \frac{1}{4}} w_5 \leq 2 \cdot \frac{1}{4} w_5$$

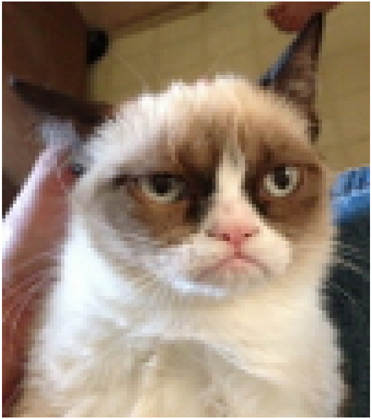
$$\frac{\partial E}{\partial z_3} = \frac{\partial L}{\partial z_4}\frac{\partial z_4}{\partial z_3} \leq 2 \cdot (\frac{1}{4})^2 w_4 w_5$$

$$\frac{\partial E}{\partial x} = \frac{\partial L}{\partial z_1}\frac{\partial z_1}{\partial x} \leq 2 \cdot (\frac{1}{4})^5 w_1 w_2 w_3 w_4 w_5$$

Backprop, паралич сети, выводы

- Значение градиента затухает экспоненциально $\Rightarrow$ сходимость замедляется
- При малых значениях весов этот эффект усиливается
- При больших значениях весов значение градиента может экспоненциально возрастать $\Rightarrow$ алгоритм расходится
- Эффект мало заметен у сетей с малым числом слоев

Проблема паралича сети, визуализация



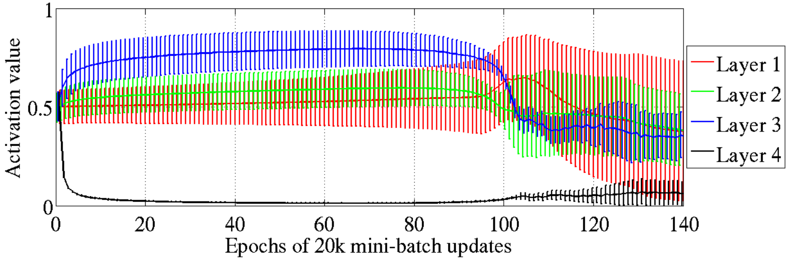
(a) Исходное изображение

(b) Образ в первом скрытом слое

Рис.: Оригинал и его образ в первом скрытом слое полносвязной нейросети при инициализации весов $w_{ij} \sim \mathcal{N}(0, 0.01)$

Сеть в процессе обучения [5]

- После случайной инициализации каждый слой получает шум, поэтому лучше всего игнорировать входы
- Сигмоида: $\sigma(z) = \frac{1}{1+e^{-z}}$
- Игнорирование входа: $\sigma(z) = 0$, для этого $z \to -\infty$



[5] http://jmlr.org/proceedings/papers/v9/glorot10a/glorot10a.pdf

Проблемы обучения глубинных нейросетей

- Vanishing/Exploding gradients
- Обученные нейроны (активация близка к 0 или 1) блокируют передачу сигнала
- Очень много параметров — высок риск переобучения

Предобработка данных и регуляризация

Предобработка данных

- Вычитание среднего
- Декорелляция данных
- Масштабирование к единичной дисперсии

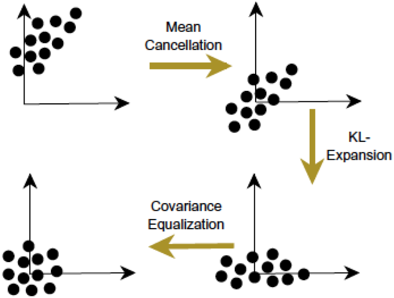


Рис.: Полный процесс предобработки[6]

[6] Efficient BackProp, Yann A. LeCun, Léon Bottou, et. al

Перемешивание примеров

- Рекомендуется перемешивать данные перед каждой эпохой[7]
- Батчи должны содержать данные как можно большего числа различных классов
- Имеет смысл чаще показывать экземпляры, на которых допускается большая ошибка. Следует быть аккуратным в присутствии выбросов

[7] эпоха - проход через весь набор данных

Регуляризация

Дополнительный штраф: $L_R = L\left(\vec{y}, \vec{t}\right) + \lambda \cdot R(W)$

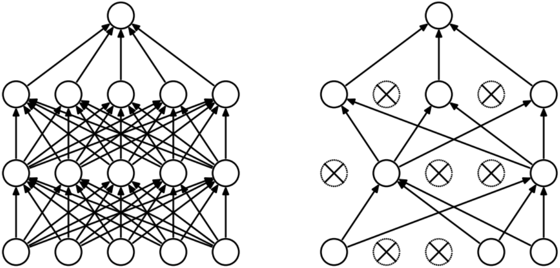
L2 регуляризация:

- $R_{L2}(W) = \dfrac{1}{2} \sum_i w_i^2$
- $\dfrac{\partial R_{L2}(W)}{\partial w_i} = w_i$

L1 регуляризация:

- $R_{L1}(W) = \sum_i |w_i|$
- $\dfrac{\partial R_{L1}(W)}{\partial w_i} = sign(w_i)$

Dropout[8]



- С вероятностью p занулим выход нейрона (рекомендовано $p = 0.4$)
- В test-time домножаем веса на вероятность сохранения
- Не стоит выкидывать нейроны последнего слоя

[8] Dropout: A Simple Way to Prevent Neural Networks from OverfittingN. Srivastava, G. Hinton

Dropout, мотивация

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

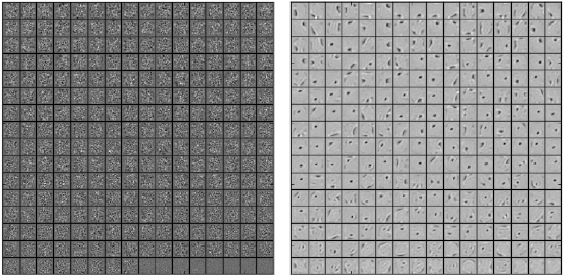


Рис.: Выученные признаки на MNIST (автокодировщик с одним скрытым слоем и ReLU в качестве активации). Слева: без Dropout, справа – с Dropout

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 [9]

- 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?

[9] https://arxiv.org/abs/1502.03167

Батч-нормализация [9]

- 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 $\Rightarrow$ линейная модель :(
- Доп. параметры: $y^{(k)} = \gamma^{(k)}\hat{x}^{(k)} + \beta^{(k)}$

[9] https://arxiv.org/abs/1502.03167

Алгоритм

Входы: Значения x в мини-батче $\mathcal{B} = \{x_{1\ldots m}\}$;
Параметры: γ, β
Выход: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// среднее мини-батча}$$

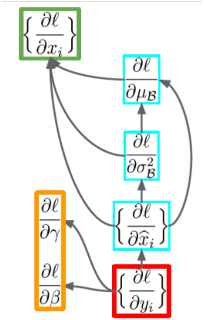
$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// дисперсия мини-батча}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// нормализация}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// растяжение и сдвиг}$$

Градиент

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



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

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

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

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

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

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

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: $E_{i+1} = (1 - \alpha)E_i + \alpha E_{\mathcal{B}}$

Tips

Стоит помнить, что с батч-нормализацией:

- Надо убрать смещения
- Увеличить темп обучения
- Уменьшить вероятность Dropout
- Уменьшить L_2 регуляризацию
- Быстрее уменьшать темп обучения
- Перемешивать обучающую выборку

Результаты

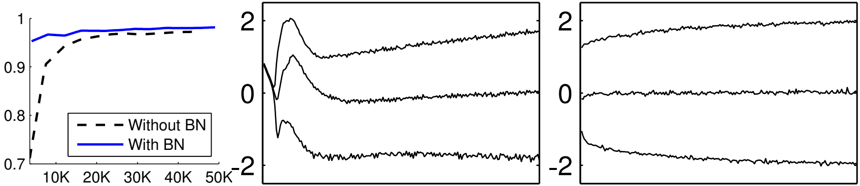


Рис.: (L): Тестовая точность на MNIST, (M): распределение входов в сигмоиду, {15, 50, 85}-е перцентили, без батч-нормализации, (R): с батч-нормализацией

Обучение

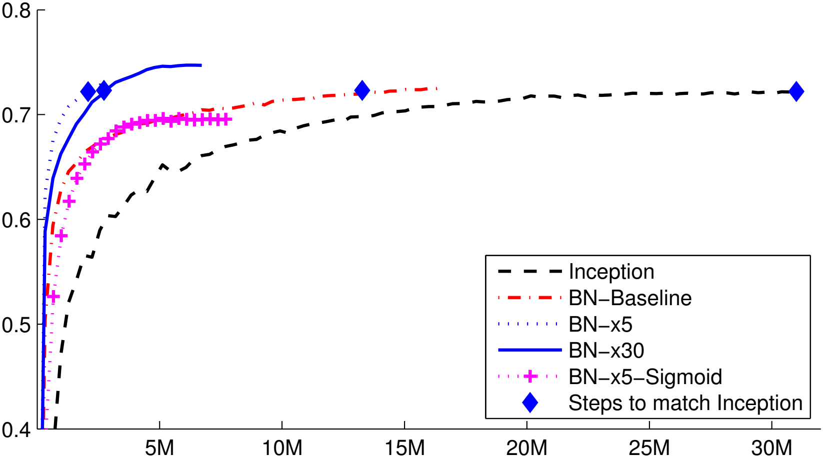


Рис.: Обучение Inception с и без батч-нормализации.[10]

[10] x30 — увеличение темпа обучения в 30 раз

Инициализация весов

Xavier (Glorot)

Рассмотрим нечетную функцию с единичной производной в нуле в качестве активации (нпр. 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 [11]

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:

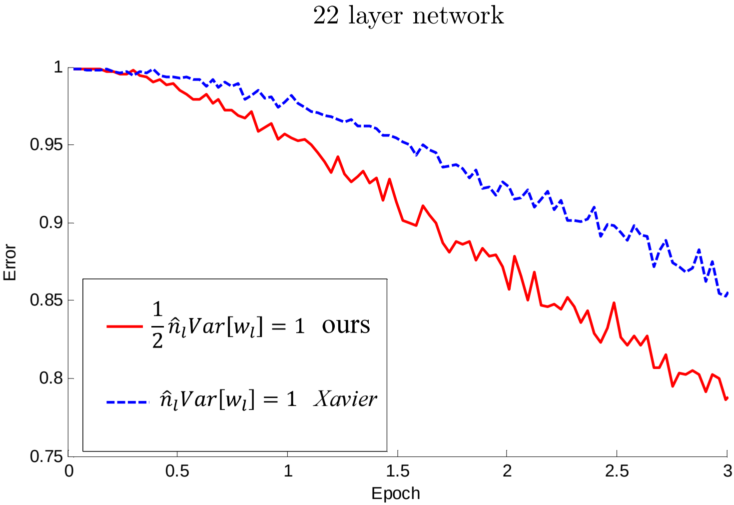
$$\mathbb{D}[\frac{\partial L}{\partial s^i}] = \mathbb{D}[\frac{\partial L}{\partial s^d}] \prod_{k=1}^{d} \frac{1}{2} n_{k+1} \mathbb{D}[W^k] = \frac{n_2}{n_d} \mathbb{D}[\frac{\partial L}{\partial s^d}]$$

n_2/n_d небольшое для сверточных сетей

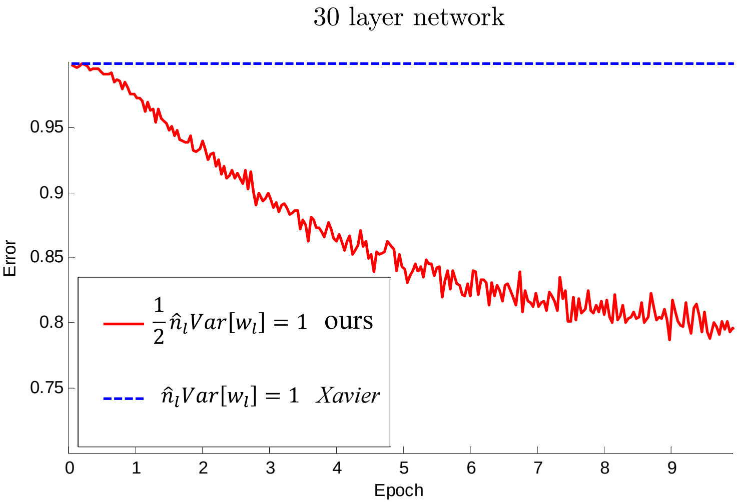
$$\boxed{\begin{array}{c} W^i \sim N(0, \frac{2}{n_i}) \\ \text{or} \\ W^i \sim N(0, \frac{2}{n_{i+1}}) \end{array}}$$

[11] https://arxiv.org/abs/1502.01852

Xavier против He для ReLU



Xavier против He для ReLU



Вопросы

1. Почему глубинные сети плохо учатся?
2. Для чего надо предобрабатывать данные?
3. Что такое dropout?
4. Что такое batch normalization?
5. Как инициализировать веса?