Глубинное обучение

Батч-нормализация. Инициализация. Эвристики для обучения нейронных сетей

Даниил Водолазский

ВШЭ

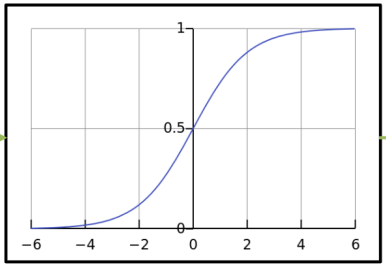
14 июля 2021 г.

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Содержание

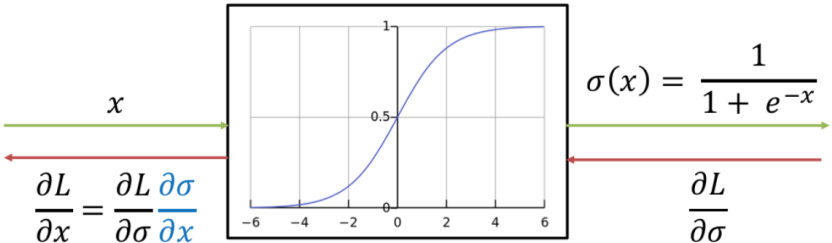
Sigmoid activation

x



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

Sigmoid activation



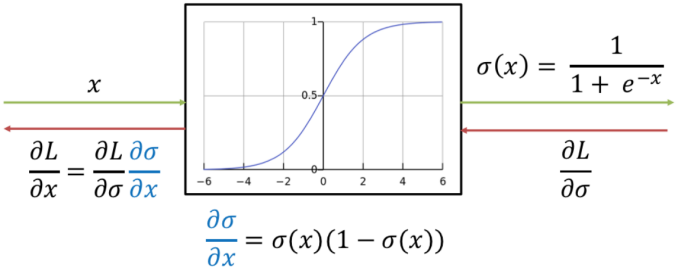
Паралич сети

- В случае сигмоиды $\sigma'(x) = \sigma(x) \cdot (1 - \sigma(x))$.
- Сигмоида выдает значения на отрезке $[0; 1]$, поэтому её производная не превосходит $\frac{1}{4}$.
- Если сеть очень глубокая, происходит затухание градиента.
- Градиент затухает экспоненциально $\Rightarrow$ сходимость замедляется, более ранние веса обновляются дольше, более глубокие веса быстрее $\Rightarrow$ значение градиента становится ещё меньше $\Rightarrow$ наступает паралич сети.
- В сетях с небольшим числом слоёв этот эффект незаметен.

Центрирование

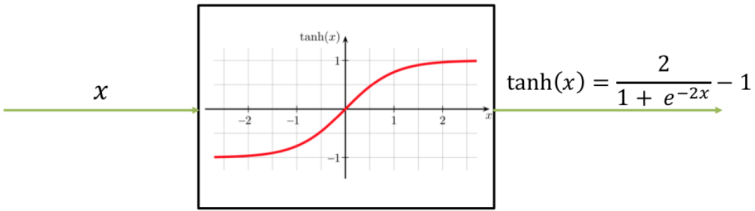
- Сигмоида не центрирована относительно нуля.
- Выход слоя мы обычно находим как $o_i = \sigma(h_i)$, он всегда положительный, поэтому градиент по весам, идущим на вход в текущий нейрон, тоже положительные $\Rightarrow$ они обновляются в одинаковом направлении.
- Сходимость идёт медленнее и зигзагообразно, но идёт

Sigmoid activation



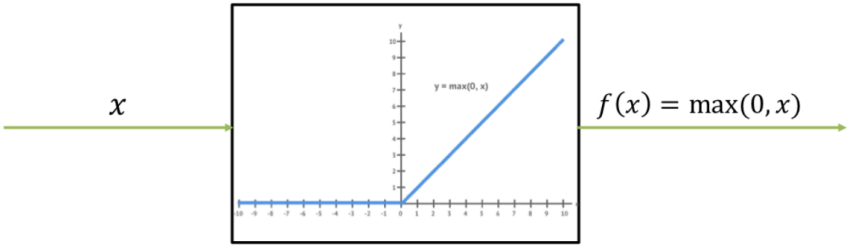
- Способствует затуханию градиента.
- Не центрирована относительно нуля.
- Вычислять e^x дорого.

Tanh activation



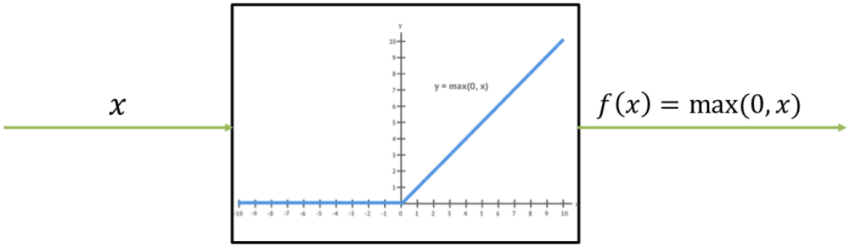
- Центрирован относительно нуля .
- Всё ещё похож на сигмоиду .
- $f'(x) = 1 - f(x)^2 \Rightarrow$ затухание градиента.

ReLU activation



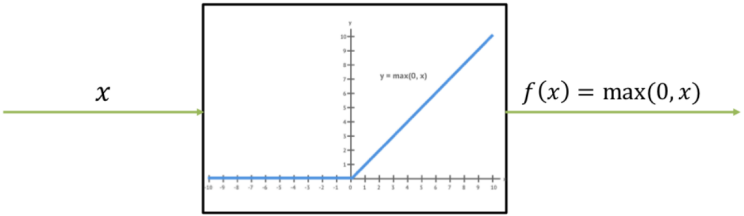
- Быстро вычисляется.
- Градиент не затухает.
- Сходимость сеток ускоряется.

ReLU activation



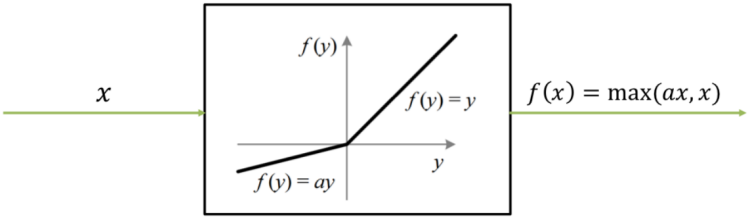
- Сетка может умереть, если активация занулится на всех нейронах.
- Не центрирован относительно нуля.

Зануление ReLU



- $f(x) = \max(0, w_0 + w_1 \cdot h_1 + ... + w_k \cdot h_k)$.
- Если w_0 инициализировано большим отрицательным числом, нейрон сразу умирает $\Rightarrow$ надо аккуратно инициализировать веса.

Leaky ReLU activation



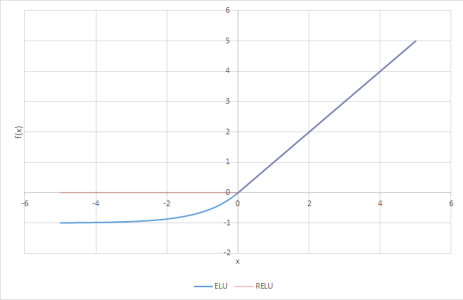
- Как ReLU, но не умирает, всё ещё легко считается.
- Производная может быть любого знака.
- Важно, чтобы $a \neq 1$, иначе линейность.

Что же выбрать

- Обычно начинают с ReLU, если сетка умирает, берут LeakyReLU.
- ReLU — стандартный выбор для свёрточных сетей.
- В рекуррентных сетях чаще всего предпочитается $tanh$, но встречаются и сигмоида, и другие активации.
- На самом деле это не очень важно, нужно держать в голове свойства функций, о которых выше шла речь, и понимать, что от перебора функций обычно выигрыш в качестве довольно низкий.
- Но есть и исключения.

Краткий обзор функций активаций: https://arxiv.org/pdf/1804.02763.pdf

ELU activation



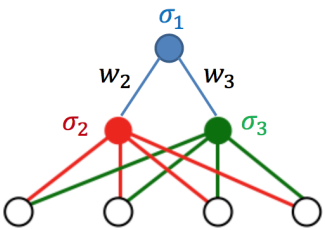
- ELU улучшает сходимость для глубоких сеток

$$f(x) = \begin{cases} x, x \geq 0, \\ \alpha \cdot (e^x - 1), x < 0. \end{cases}$$

https://arxiv.org/pdf/1511.07289.pdf

Содержание

Инициализация весов

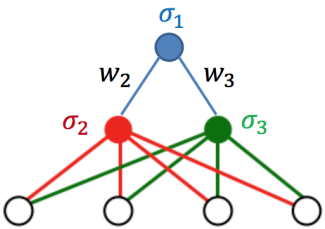


$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial \sigma_1}\sigma_1(1-\sigma_1)\sigma_2$$

$$\frac{\partial L}{\partial w_3} = \frac{\partial L}{\partial \sigma_1}\sigma_1(1-\sigma_1)\sigma_3$$

- Что будет, если инициализировать веса нулями?

Инициализация весов

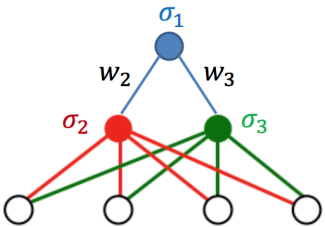


$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial \sigma_1}\sigma_1(1-\sigma_1)\sigma_2$$

$$\frac{\partial L}{\partial w_3} = \frac{\partial L}{\partial \sigma_1}\sigma_1(1-\sigma_1)\sigma_3$$

- Что будет, если инициализировать веса нулями?
- σ_2 и σ_3 будут обновляться одинаково.

Инициализация весов

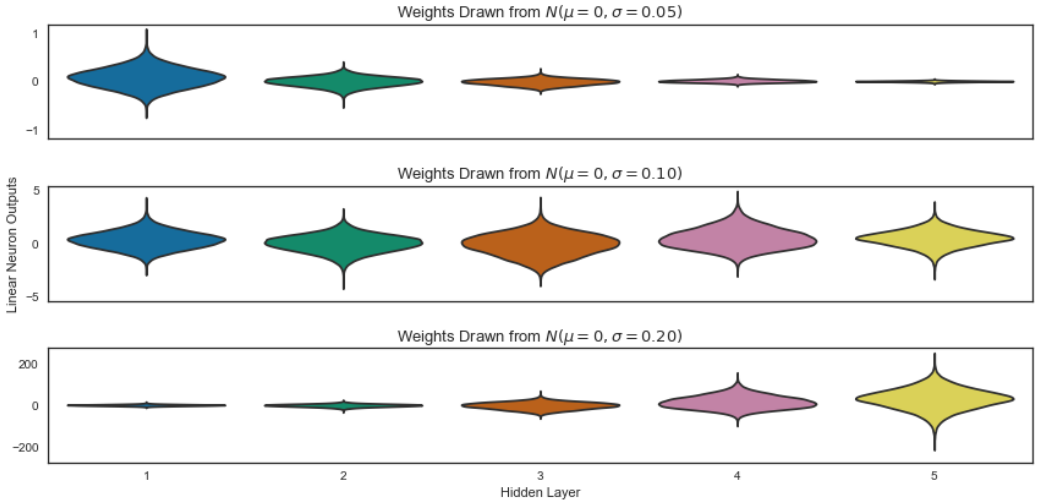


$$\frac{\partial L}{\partial w_2} = \frac{\partial L}{\partial \sigma_1}\sigma_1(1-\sigma_1)\sigma_2$$

$$\frac{\partial L}{\partial w_3} = \frac{\partial L}{\partial \sigma_1}\sigma_1(1-\sigma_1)\sigma_3$$

- Хочется уничтожить симметрию.
- Обычно инициализируют маленькими случайными числами из какого-то распределения (нормальное, равномерное).

Симметричный случай



Инициализация весов

- Наши признаки X пришли к нам из какого-то распределения.
- Выход слоя $f(XW)$ будет принадлежать другому распределению.
- Если инициализировать веса неправильно, дисперсия распределения може от слоя к слою затухать (сигнал будет теряться) либо наоборот, возрастать (сигнал будет рассеиваться).
- Эмпирически было выяснено, что это может портить сходимость для глубоких сеток.
- Хочется контролировать дисперсию.

Инициализация весов

- Посмотрим на выход нейрона перед активацией:

$$h_i = w_0 + \sum_{i=1}^{n_{in}} w_i x_i.$$

- Дисперсия h_i выражается через дисперсии x и w.
- Она не зависит от константы w_0.
- Будем считать, что веса $w_1, \dots, w_k \sim iid$, наблюдения $x_1, \dots, x_n \sim iid$, а ещё x_i и w_i независимы между собой.

Инициализация весов (симметричный случай)

$$\mathrm{Var}(h_i) = \mathrm{Var}(\sum_{i=1}^{n_{in}} w_i x_i) = \sum_{i=1}^{n_{in}} \mathrm{Var}(w_i x_i) =$$

$$= \sum_{i=1}^{n_{in}} [\mathbb{E}(x_i)]^2 \cdot \mathrm{Var}(w_i) + [\mathbb{E}(w_i)]^2 \cdot \mathrm{Var}(x_i) + \mathrm{Var}(x_i) \cdot \mathrm{Var}(w_i)] =$$

Инициализация весов (симметричный случай)

$$\mathrm{Var}(h_i) = \mathrm{Var}(\sum_{i=1}^{n_{in}} w_i x_i) = \sum_{i=1}^{n_{in}} \mathrm{Var}(w_i x_i) =$$

$$= \sum_{i=1}^{n_{in}} [\mathbb{E}(x_i)]^2 \cdot \mathrm{Var}(w_i) + [\mathbb{E}(w_i)]^2 \cdot \mathrm{Var}(x_i) + \mathrm{Var}(x_i) \cdot \mathrm{Var}(w_i)] =$$

- Если функция активации симметричная, тогда $\mathbb{E}(x_i) = 0$. Будем инициализировать веса с нулевым средним, тогда $\mathbb{E}(w_i) = 0$.

Инициализация весов (симметричный случай)

$$\mathrm{Var}(h_i) = \mathrm{Var}(\sum_{i=1}^{n_{in}} w_i x_i) = \sum_{i=1}^{n_{in}} \mathrm{Var}(w_i x_i) =$$

$$= \sum_{i=1}^{n_{in}} [\mathbb{E}(x_i)]^2 \cdot \mathrm{Var}(w_i) + [\mathbb{E}(w_i)]^2 \cdot \mathrm{Var}(x_i) + \mathrm{Var}(x_i) \cdot \mathrm{Var}(w_i)] =$$

$$= \sum_{i=1}^{n_{in}} \mathrm{Var}(x_i) \cdot \mathrm{Var}(w_i)$$

- Если функция активации симметричная, тогда $\mathbb{E}(x_i) = 0$. Будем инициализировать веса с нулевым средним, тогда $\mathbb{E}(w_i) = 0$.

Инициализация весов (симметричный случай)

$$\mathrm{Var}(h_i) = \mathrm{Var}(\sum_{i=1}^{n_{in}} w_i x_i) = \sum_{i=1}^{n_{in}} \mathrm{Var}(w_i x_i) =$$

$$= \sum_{i=1}^{n_{in}} [\mathbb{E}(x_i)]^2 \cdot \mathrm{Var}(w_i) + [\mathbb{E}(w_i)]^2 \cdot \mathrm{Var}(x_i) + \mathrm{Var}(x_i) \cdot \mathrm{Var}(w_i)] =$$

$$= \sum_{i=1}^{n_{in}} \mathrm{Var}(x_i) \cdot \mathrm{Var}(w_i) = \mathrm{Var}(x) \cdot [n_{in} \cdot \mathrm{Var}(w)]$$

Инициализация весов (симметричный случай)

$$\mathrm{Var}(h_i) = \mathrm{Var}(\sum_{i=1}^{n_{in}} w_i x_i) = \sum_{i=1}^{n_{in}} \mathrm{Var}(w_i x_i) =$$

$$= \sum_{i=1}^{n_{in}} [\mathbb{E}(x_i)]^2 \cdot \mathrm{Var}(w_i) + [\mathbb{E}(w_i)]^2 \cdot \mathrm{Var}(x_i) + \mathrm{Var}(x_i) \cdot \mathrm{Var}(w_i)] =$$

$$= \sum_{i=1}^{n_{in}} \mathrm{Var}(x_i) \cdot \mathrm{Var}(w_i) = \mathrm{Var}(x) \cdot \underbrace{[n_{in} \cdot \mathrm{Var}(w)]}_{=1}$$

Плохая инициализация весов

Пусть

$$w_i \sim U\left[-\frac{1}{\sqrt{n_{in}}}; \frac{1}{\sqrt{n_{in}}}\right],$$

тогда

$$\mathrm{Var}(w_i) = \frac{1}{12} \cdot \left(\frac{1}{\sqrt{n_{in}}} + \frac{1}{\sqrt{n_{in}}}\right)^2 = \frac{1}{3n_{in}} \Rightarrow Var(h_i) = \frac{1}{3}$$

Получаем затухание!

Немного лучше

Пущай

$$w_i \sim U\left[-\frac{\sqrt{3}}{\sqrt{n_{in}}}; \frac{\sqrt{3}}{\sqrt{n_{in}}}\right],$$

тогда

$$\mathrm{Var}(w_i) = \frac{1}{12} \cdot \left(\frac{\sqrt{3}}{\sqrt{n_{in}}} + \frac{\sqrt{3}}{\sqrt{n_{in}}}\right)^2 = \frac{1}{n_{in}} \Rightarrow Var(h_i) = 1$$

Немного лучше

Пущай

$$w_i \sim U\left[-\frac{\sqrt{3}}{\sqrt{n_{in}}}; \frac{\sqrt{3}}{\sqrt{n_{in}}}\right],$$

тогда

$$\mathrm{Var}(w_i) = \frac{1}{12} \cdot \left(\frac{\sqrt{3}}{\sqrt{n_{in}}} + \frac{\sqrt{3}}{\sqrt{n_{in}}}\right)^2 = \frac{1}{n_{in}} \Rightarrow Var(h_i) = 1$$

При forward pass на вход идёт n_{in} набобдений, при backward pass на вход идёт n_{out} градиентов $\Rightarrow$ канал с дисперсией может быть непостоянным, если число весов от слоя к слою сильно колеблется

Инициализация Ксавье (Глоро)

Для неодинаковых размеров слоёв невозможно удволетворить обоим условиям, поэтому обычно усредняют:

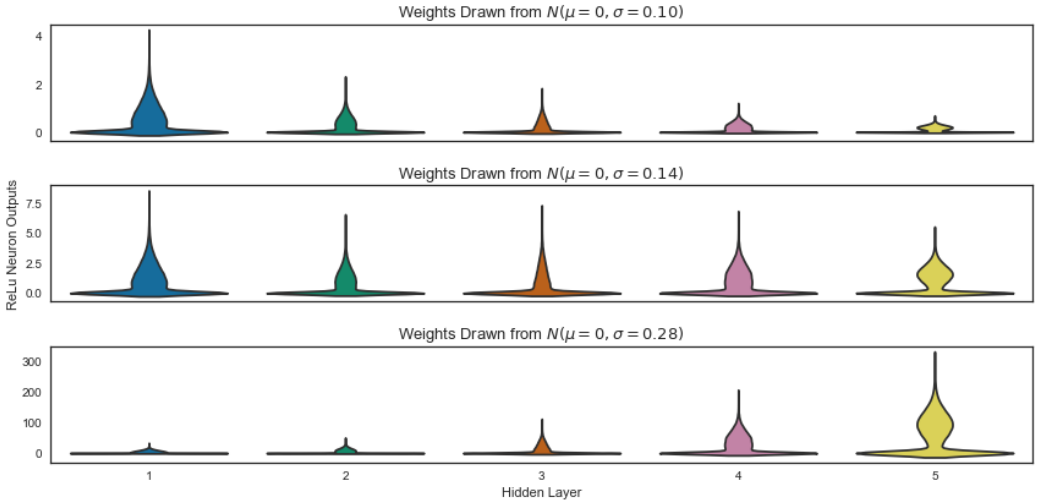
$$w_i \sim U\left[-\frac{\sqrt{6}}{\sqrt{n_{out}+n_{in}}}; \frac{\sqrt{6}}{\sqrt{n_{out}+n_{in}}}\right],$$

Такая инициализация называется инициализацией Ксавие (или Глоро)

По аналогии можно найти формулу для дисперсии нормального распределения, но это уже семинарская задачка :)

http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf

Несимметричный случай



Инициализация Xe

$$\mathrm{Var}(h_i) = \mathrm{Var}(\sum_{i=1}^{n_{in}} w_i x_i) = \sum_{i=1}^{n_{in}} \mathrm{Var}(w_i x_i) =$$

$$= \sum_{i=1}^{n_{in}} [\mathbb{E}(x_i)]^2 \cdot \mathrm{Var}(w_i) + [\mathbb{E}(w_i)]^2 \cdot \mathrm{Var}(x_i) + \mathrm{Var}(x_i) \cdot \mathrm{Var}(w_i)]$$

- Когда нет симметрии, можно занулить только второе слагаемое

Инициализация Xe

$$\mathrm{Var}(h_i) = \mathrm{Var}(\sum_{i=1}^{n_{in}} w_i x_i) = \sum_{i=1}^{n_{in}} \mathrm{Var}(w_i x_i) =$$

$$= \sum_{i=1}^{n_{in}} [\mathbb{E}(x_i)]^2 \cdot \mathrm{Var}(w_i) + [\mathbb{E}(w_i)]^2 \cdot \mathrm{Var}(x_i) + \mathrm{Var}(x_i) \cdot \mathrm{Var}(w_i)] =$$

$$= \sum_{i=1}^{n_{in}} [\mathbb{E}(x_i)]^2 \cdot \mathrm{Var}(w_i) + \mathrm{Var}(x_i) \cdot \mathrm{Var}(w_i) = \sum_{i=1}^{n_{in}} \mathrm{Var}(w_i) \cdot \mathbb{E}(x_i^2)$$

- Когда нет симметрии, можно занулить только второе слагаемое

Инициализация Xe

$$\mathrm{Var}(h_i) = \mathrm{Var}(\sum_{i=1}^{n_{in}} w_i x_i) = \sum_{i=1}^{n_{in}} \mathrm{Var}(w_i x_i) =$$

$$= \sum_{i=1}^{n_{in}} [\mathbb{E}(x_i)]^2 \cdot \mathrm{Var}(w_i) + [\mathbb{E}(w_i)]^2 \cdot \mathrm{Var}(x_i) + \mathrm{Var}(x_i) \cdot \mathrm{Var}(w_i)] =$$

$$= \sum_{i=1}^{n_{in}} [\mathbb{E}(x_i)]^2 \cdot \mathrm{Var}(w_i) + \mathrm{Var}(x_i) \cdot \mathrm{Var}(w_i) = \sum_{i=1}^{n_{in}} \mathrm{Var}(w_i) \cdot \mathbb{E}(x_i^2) =$$

$$= \mathbb{E}(x^2) \cdot [n_{in} \cdot \mathrm{Var}(w)]$$

Инициализация Xe

$$\mathrm{Var}(h_i) = \mathbb{E}(x_i^2) \cdot [n_{in} \cdot \mathrm{Var}(w)]$$
$$x_i = \max(0; h_{i-1})$$

Инициализация Xe

$$\mathrm{Var}(h_i) = \mathbb{E}(x_i^2) \cdot [n_{in} \cdot \mathrm{Var}(w)]$$
$$x_i = \max(0; h_{i-1})$$

Если h_{i-1} симметрично распределён относительно нуля, тогда:

$$\mathbb{E}(x_i^2) = \frac{1}{2} \cdot \mathrm{Var}(h_{i-1})$$

https://arxiv.org/pdf/1502.01852.pdf

Инициализация Xe

$$\mathrm{Var}(h_i) = \mathbb{E}(x_i^2) \cdot [n_{in} \cdot \mathrm{Var}(w)]$$
$$x_i = \max(0; h_{i-1})$$

Если h_{i-1} симметрично распределён относительно нуля, тогда:

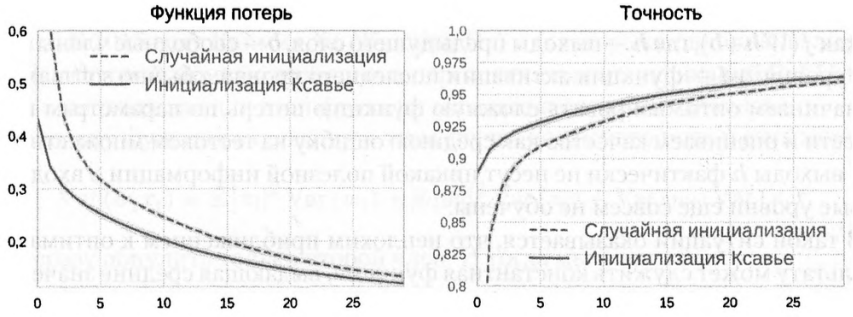
$$\mathbb{E}(x_i^2) = \frac{1}{2} \cdot \mathrm{Var}(h_{i-1})$$
$$\mathrm{Var}(h_i) = \frac{1}{2} \cdot \mathrm{Var}(h_{i-1}) \cdot [n_{in} \cdot \mathrm{Var}(w)]$$
$$\mathrm{Var}(w_i) = \frac{2}{n_{in}}$$

https://arxiv.org/pdf/1502.01852.pdf

Кратикие итоги

- Для симметричных функций с нулевым средним используйте инициализацию Ксавье.
- Для ReLU и им подобным инициализацию Хе.
- Эти две инициализации корректируют параметры распределений в зависимости от входа и выхода слоя так, чтобы поддерживать дисперсию равной единице.
- `https://pytorch.org/docs/stable/nn.init.html`

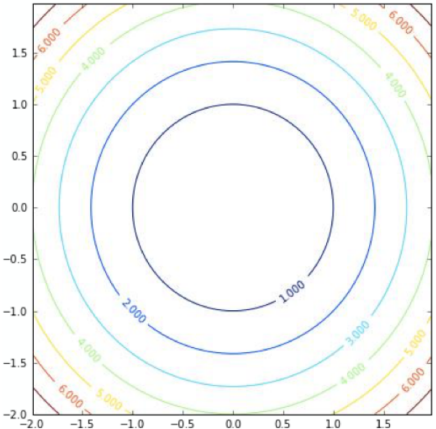
Эксперимент с MNIST

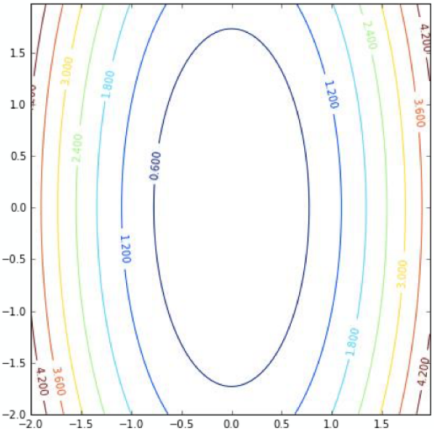


Источник: Николенко, страница 149

Содержание

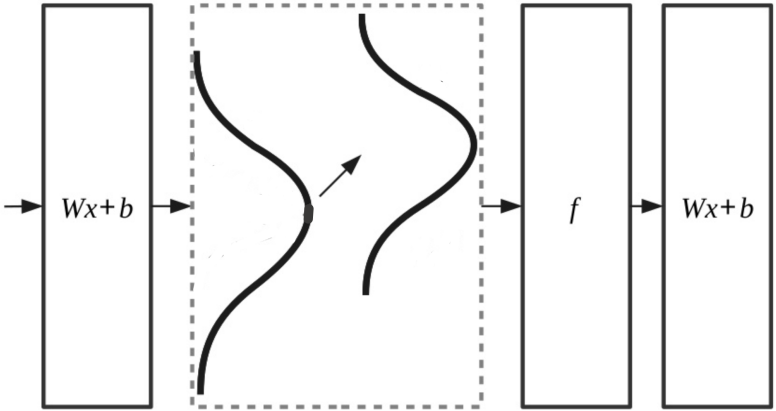
Стандартизация



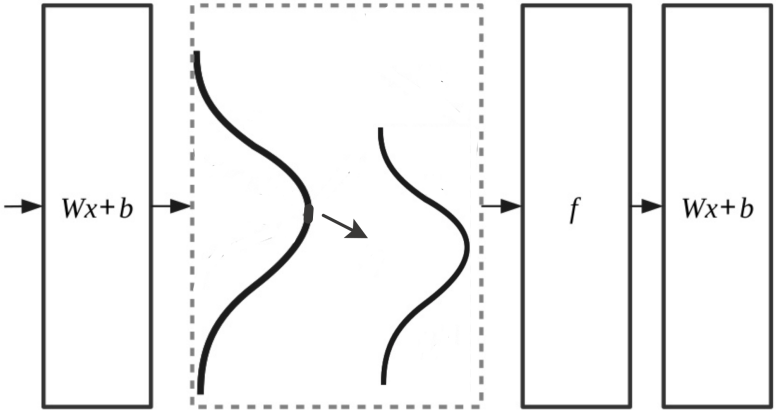


Какая из ситуаций лучше для SGD?

А что внутри?



А что внутри?



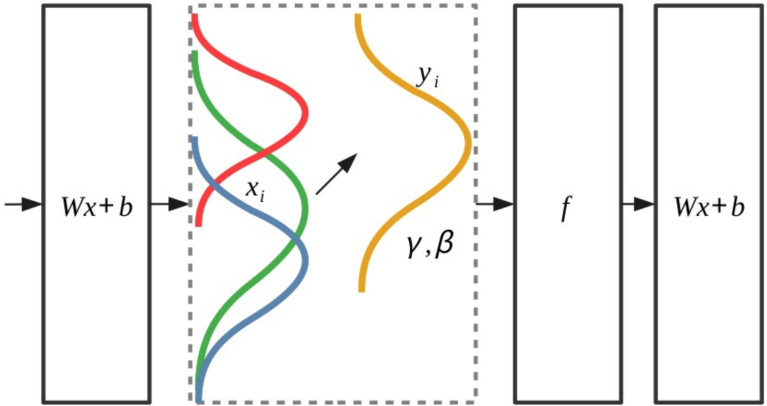
Проблема

- Давайте вместо X на входе использовать $\frac{X-\mu_X}{\sigma_X}$.
- Даже если мы стандартизовали вход X, внутри сетки может произойти несчастье и скрытый слой окажется не стандартизован.
- Скрытые представления $h = f(XW)$ могут менять своё распределение в процессе обучения, это усложняет его.

Проблема

- Давайте вместо X на входе использовать $\frac{X-\mu_X}{\sigma_X}$.
- Даже если мы стандартизовали вход X, внутри сетки может произойти несчастье и скрытый слой окажется не стандартизован.
- Скрытые представления $h = f(XW)$ могут менять своё распределение в процессе обучения, это усложняет его.
- Давайте на каждом слое вместо h использовать $\hat{h} = \frac{h-\mu_h}{\sigma_h}$.
- На выход будем выдавать $\beta \cdot \hat{h} + \gamma$, для того, чтобы у нас было больше свободы; параметры β и γ тоже учим.

Batch norm (2015)



Batch norm (2015)

- Откуда взять μ_h и σ_h?
- Оценить по текущему батчу B!

$$\mu_h = \alpha \cdot \mu_B + (1 - \alpha) \cdot \mu_h$$
$$\sigma_h^2 = \alpha \cdot \sigma_B^2 + (1 - \alpha) \cdot \sigma_h^2$$

- Коэффициенты β и γ оцениваются в ходе обратного распространения ошибки.
- Обучение довольно сильно ускоряется, сходимость улучшается.

https://arxiv.org/pdf/1502.03167.pdf

Forward pass

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

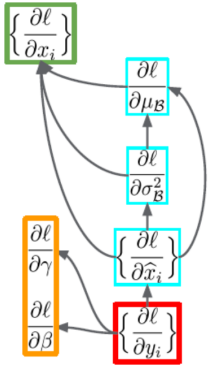
$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

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

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

https://arxiv.org/pdf/1502.03167.pdf

Backward pass



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

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

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

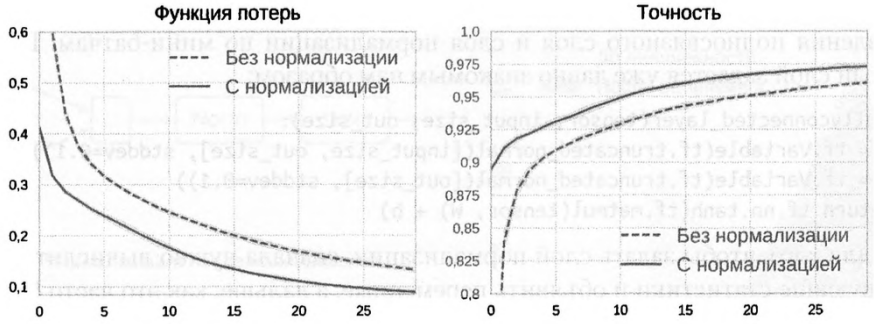
$$\frac{\partial \ell}{\partial x_i} = \frac{\partial \ell}{\partial \widehat{x}_i} \cdot \frac{1}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} + \frac{\partial \ell}{\partial \sigma^2_{\mathcal{B}}} \cdot \frac{2(x_i - \mu_{\mathcal{B}})}{m-1} + \frac{\partial \ell}{\partial \mu_{\mathcal{B}}} \cdot \frac{1}{m}$$

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

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

https://arxiv.org/pdf/1502.03167.pdf

Эксперимент с MNIST

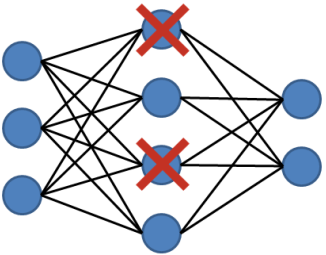


Источник: Николенко, страница 160

Содержание

Дропаут

- С вероятностью p отключаем нейрон.
- Делает нейроны более устойчивыми к случайным возмущениям.
- Борьба с коадоптацией: не все соседи похожи, не все дети похожи на родителей.



http://www.cs.toronto.edu/~rsalakhu/papers/srivastava14a.pdf

Dropout в формулах

- Forward pass:

$$h = f(XW + b),$$
$$o = D \cdot h,$$

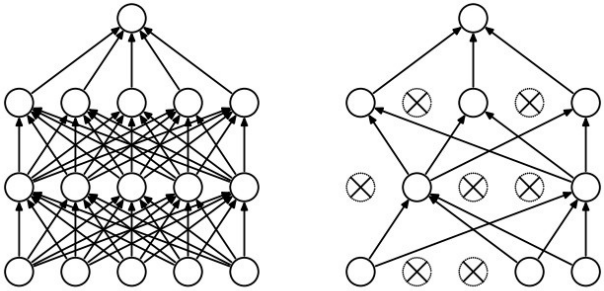
где $D = (D_1, \dots, D_k) \sim Bernoulli(1-p)\,(i.i.d.)$.
Дропаут — это просто небольшая модификация функции активации.

- Backward pass:

$$\frac{\partial L}{\partial o} = D \cdot \frac{\partial L}{\partial h}.$$

Dropout

- При обучении мы домножаем часть выходов на D_i, тем самым мы изменяем только часть параметров и нейроны учатся более независимо.
- Dropout эквивалентен обучению 2^n сетей.



(a) Standard Neural Net

(b) After applying dropout.

Dropout

- При обучении мы домножаем часть выходов на D_i, тем самым мы изменяем только часть параметров и нейроны учатся более независимо.
- Dropout эквивалентен обучению 2^n сетей.
- Что делать на стадии тестирования?

Dropout

- При обучении мы домножаем часть выходов на D_i, тем самым мы изменяем только часть параметров и нейроны учатся более независимо.
- Dropout эквивалентен обучению 2^n сетей.
- Нам надо сымитировать работу такого ансамбля: можно отключать по очереди все возможные комбинации нейронов, получить 2^n прогнозов и усреднить их.

Dropout

- При обучении мы домножаем часть выходов на D_i, тем самым мы изменяем только часть параметров и нейроны учатся более независимо.
- Dropout эквивалентен обучению 2^n сетей.
- Нам надо сымитировать работу такого ансамбля: можно отключать по очереди все возможные комбинации нейронов, получить 2^n прогнозов и усреднить их.
- Но лучше просто брать по дропауту математическое ожидание

$$o = (1 - p) \cdot f(X \cdot W + b).$$

Обратный Dropout

- На тесте ищем математическое ожидание:

$$o = (1 - p) \cdot f(X \cdot W + b).$$

Обратный Dropout

- На тесте ищем математическое ожидание:

$$o = (1 - p) \cdot f(X \cdot W + b).$$

- Это неудобно! Надо переписывать функцию для прогнозов!

Обратный Dropout

- На тесте ищем математическое ожидание:

$$o = (1 - p) \cdot f(X \cdot W + b).$$

- Это неудобно! Надо переписывать функцию для прогнозов!
- Давайте лучше будем домножать на $\frac{1}{1-p}$ на этапе обучения:

$$\text{train: } o = \frac{1}{1-p} \cdot D \cdot f(X \cdot W + b),$$
$$\text{test: } o = f(X \cdot W + b).$$

Обобщения

- Монте-карловский дропаут. https://arxiv.org/abs/1506.02142v6

Содержание

Предобучение

- На будущее: обучаем на корпусе картинок автокодировщик, encoder благодаря этому учится выделять наиболее важные признаки, которые позволяют эффективно сжимать изображения. После срезаем decoder и на его месте достраиваем слои для решения нашей задаче, запускаем обычное дообучение.
- Успешно применяется на практике в обработке изображений и текстов, так как обучать собственные огромные модели многим не по карману, да и обычно незачем.

Динамическое наращивание сети

1. Обучение сети при заведомо недостаточном числе нейронов.
2. После стабилизации функции потерь — добавление нового нейрона и его инициализация путём обучения
   - либо по случайной подвыборке;
   - либо по объектам с наибольшими значениями потерь;
   - либо по случайному подмножеству входов;
   - либо из различных случайных начальных приближений.
3. Снова итерации BackProp.

Эмпирический опыт: общее время обучения обычно лишь в $1.5 - 2$ раза больше, чем если бы в сети сразу было итоговое число нейронов. Полезная информация, накопленная сетью, не теряется при добавлении нейронов.

Прореживание сети

1. Начать с большого количество нейронов и удалять незначимые по какому-нибудь критерию.
2. После прореживания снова запускаем backprop.
3. Если качество модели сильно упала, надо вернуть последние удалённые связи.

Пример: обнуляем веса; смотрим, как сильно упала ошибка; сортируем все связи по этому критерию, удаляем N наименее значимых.

Содержание

Другие хаки

- Ранняя остановка.
- L_1- и L_2-регуляризация.
- Различные новые градиентные спуски, ускоряющие процедуру сходимости.
- Skip connections
- Аугментация данных
- Более забубуенистые архитектуры

Ранняя остановка

- Будем останавливать обучение, когда качество на валидации начинает падать или перестает расти.
- Критерием может служить как функция потерь, так и метрики.

Регуляризация

- L_2: приплюсовываем к функции потерь $\frac{\lambda}{2}||W||_2^2 = \sum_{w \in W} |w|^2$.
- L_1: приплюсовываем к функции потерь $\mu||W||_1^1 = \mu \sum_{w \in W} |w|$.
- Можно регуляризовать не всю сетку, а отдельный нейрон или слой. В PyTorch у оптимизаторов есть параметр `weight_decay`, он применяется ко всем весам. Если хотите навесить регуляризацию на некоторые параметры, нужно добавить ее к лоссу вручную.
- Не даёт нейрону сфокусироваться на слишком выделяющемся входе.
- Помогает от переобучения.

Взаимосвязи

- На практике обычно используют дропаутом. Действия всех этих регуляризаторов оказывается схожим.

Цитата

We show that the dropout regularizer is first-order equivalent to an L2 regularizer applied after scaling the features by an estimate of the inverse diagonal Fisher information matrix.

— https://arxiv.org/abs/1307.1493

- У Гудфеллоу в Глубоком обучении на стр. 218 можно найти, что ранняя остановка для линейных моделей эквивалентна L_2-регуляризации с MSE, обучаемой SGD.

Содержание

Что узнали

- В нейронных сетях существует много мест, где можно что-то улучшить.
- И существует множество способов это сделать. основные: удачно взять функцию активации, подобрать инициализацию, добавить батч-нормализацию и / или дропаут.
- Можно сделать более продвинутое обучение, добавив регуляризацию или раннюю остановку.