OTUS

ОНЛАЙН-ОБРАЗОВАНИЕ

Взрыв и затухание градиентов

Учимся учить нейронные сети

Артур Кадурин
Преподаватель

План на сегодня

1. **Обратное распространение и градиенты**
2. Насыщение функций активации
3. Простые методы борьбы
4. Инициализация весов
5. Практика

Повторяем градиентный спуск

$$\mathcal{L}(t, y) = -\frac{1}{N}\sum_{i}[t_i \log y_i + \ (1 - t_i) \log(1 - y_i)]$$

Повторяем градиентный спуск

$$\mathcal{L}(t, y) = -\frac{1}{N}\sum_{i}[t_i \log y_i + \ (1 - t_i) \log(1 - y_i)]$$

$$y = \sigma_1\left(z_1\left(\sigma_0\left(z_0(x)\right)\right)\right)$$

$$z_i(\sigma_{i-1}) = W_i\sigma_{i-1} + b_i$$

$$\sigma_i\ (z_i) = \frac{1}{1 + e^{-z_i}}$$

Повторяем градиентный спуск

$$\mathcal{L}(t, y) = -\frac{1}{N}\sum_{i}[t_i \log y_i + (1 - t_i)\log(1 - y_i)]$$

$$y = \sigma_1\left(z_1\left(\sigma_0\left(z_0(x)\right)\right)\right)$$

$$\frac{\partial \mathcal{L}}{\partial w_0} = \frac{\partial \mathcal{L}}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial z_1}\frac{\partial z_1}{\partial \sigma_0}\frac{\partial \sigma_0}{\partial z_0}\frac{\partial z_0}{\partial w_0}$$

$$z_i(\sigma_{i-1}) = W_i\sigma_{i-1} + b_i$$

$$\sigma_i\,(z_i) = \frac{1}{1 + e^{-z_i}}$$

Повторяем градиентный спуск

$$\mathcal{L}(t, y) = -\frac{1}{N}\sum_{i}[t_i \log y_i + (1 - t_i)\log(1 - y_i)]$$

$$y = \sigma_1\left(z_1\left(\sigma_0\left(z_0(x)\right)\right)\right)$$

$$z_i(\sigma_{i-1}) = W_i\sigma_{i-1} + b_i$$

$$\sigma_i(z_i) = \frac{1}{1 + e^{-z_i}}$$

$$\frac{\partial \mathcal{L}}{\partial w_0} = \frac{\partial \mathcal{L}}{\partial \sigma_1}\frac{\partial \sigma_1}{\partial z_1}\frac{\partial z_1}{\partial \sigma_0}\frac{\partial \sigma_0}{\partial z_0}\frac{\partial z_0}{\partial w_0}$$

$$\frac{\partial \sigma_1}{\partial z_1} = \sigma_1(z_1)(1 - \sigma_1(z_1))$$

$$\frac{\partial z_1}{\partial \sigma_0} = W_1$$

Повторяем градиентный спуск

$$\mathcal{L}(t, y) = -\frac{1}{N}\sum_{i}[t_i \log y_i + \ (1 - t_i) \log(1 - y_i)]$$

$$\frac{\partial \mathcal{L}}{\partial w_0} = \frac{\partial \mathcal{L}}{\partial \sigma_N} \prod_{i=1}^{N} \left(\frac{\partial \sigma_i}{\partial z_i} \frac{\partial z_i}{\partial \sigma_{i-1}} \right) \frac{\partial \sigma_0}{\partial z_0} \frac{\partial z_0}{\partial w_0}$$

План на сегодня

1. Обратное распространение и градиенты
2. **Насыщение функций активации**
3. Простые методы борьбы
4. Инициализация весов
5. Практика

Насыщение функций активации

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

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

Насыщение функций активации

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

$$\frac{\partial tanh}{\partial z} = 1 - tanh^2(z)$$

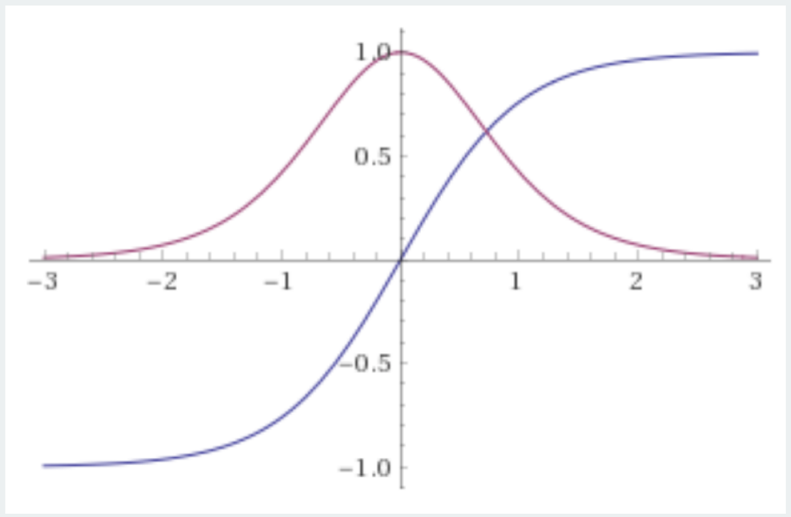
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

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

Насыщение функций активации

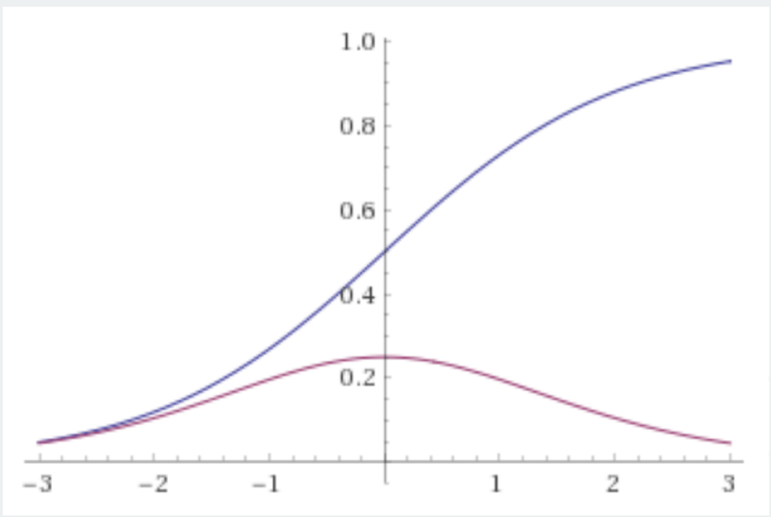
$$tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

$$\frac{\partial tanh}{\partial z} = 1 - tanh^2(z)$$



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

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



План на сегодня

1. Обратное распространение и градиенты
2. Насыщение функций активации
3. **Простые методы борьбы**
4. Инициализация весов
5. Практика

Простые методы борьбы

1. Gradient clipping

Искусственно ограничиваем градиент сверху. Часто используется при обучении рекуррентных сетей. По сути, мы соглашаемся идти в заданном направлении, но гарантировано мелкими шажками.

Простые методы борьбы

1. Gradient clipping
2. L2-регуляризация

Позволяет не только ограничить значения весов сверху, но и избежать насыщения функций активаций

Простые методы борьбы

1. Gradient clipping
2. L2-регуляризация
3. ReLU

Производная ReLU равна…

Простые методы борьбы

1. Gradient clipping
2. L2-регуляризация
3. ReLU

Производная ReLU равна 0 или 1. То есть, градиент либо не проходит, либо проходит целиком.

Простые методы борьбы

1. Gradient clipping
2. L2-регуляризация
3. ReLU
4. Инициализация весов!

План на сегодня

1. Обратное распространение и градиенты
2. Насыщение функций активации
3. Простые методы борьбы
4. **Инициализация весов**
5. Практика

Одна очень важная статья

Glorot, X., Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks.

Далее я буду использовать графики из этой статьи

Вывод первый

Так как изначально глубокая сеть по сути выдает случайные значения на выходе, простейшим решением будет…

$$y = softmax(W\boldsymbol{h} + \boldsymbol{b})$$

Вывод первый

Так как изначально глубокая сеть по сути выдает случайные значения на выходе, простейшим решением будет…

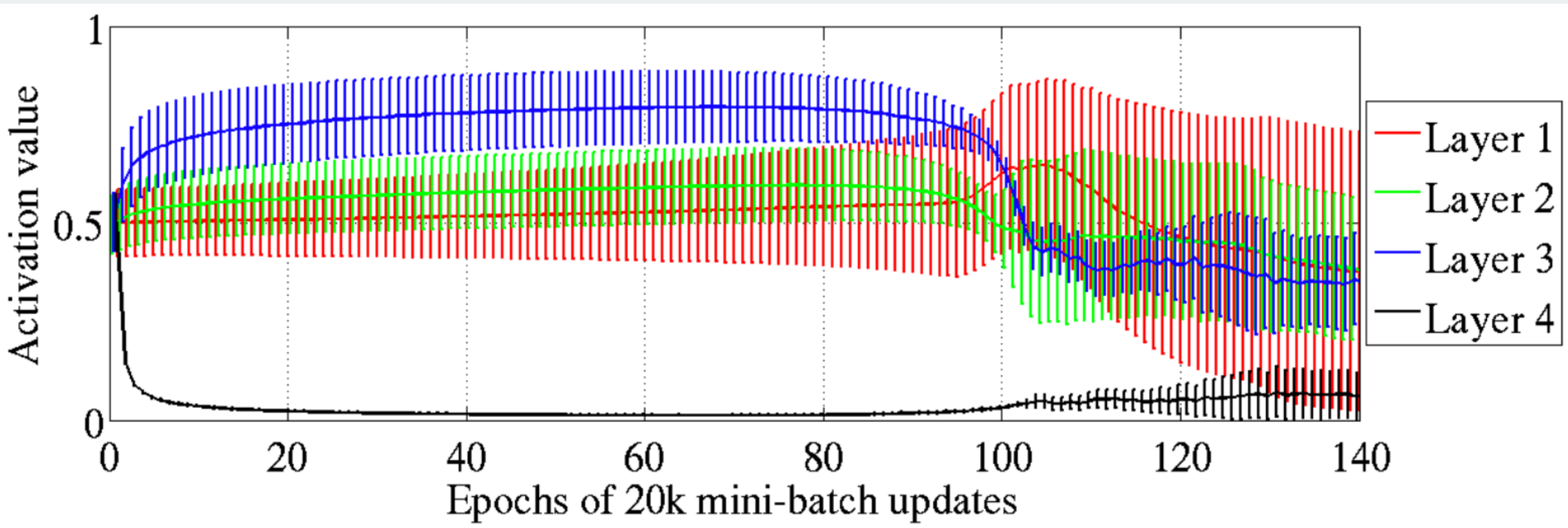
$$y = softmax(W\boldsymbol{h} + \boldsymbol{b})$$

$$\boldsymbol{b} \to \widehat{\boldsymbol{b}}, W\boldsymbol{h} \to \mathbf{0}$$

Обнуляя вклад последнего слоя мы по сути минимизируем шум, а лучшим решением будет какой-то конкретный набор байесов.

Вывод первый

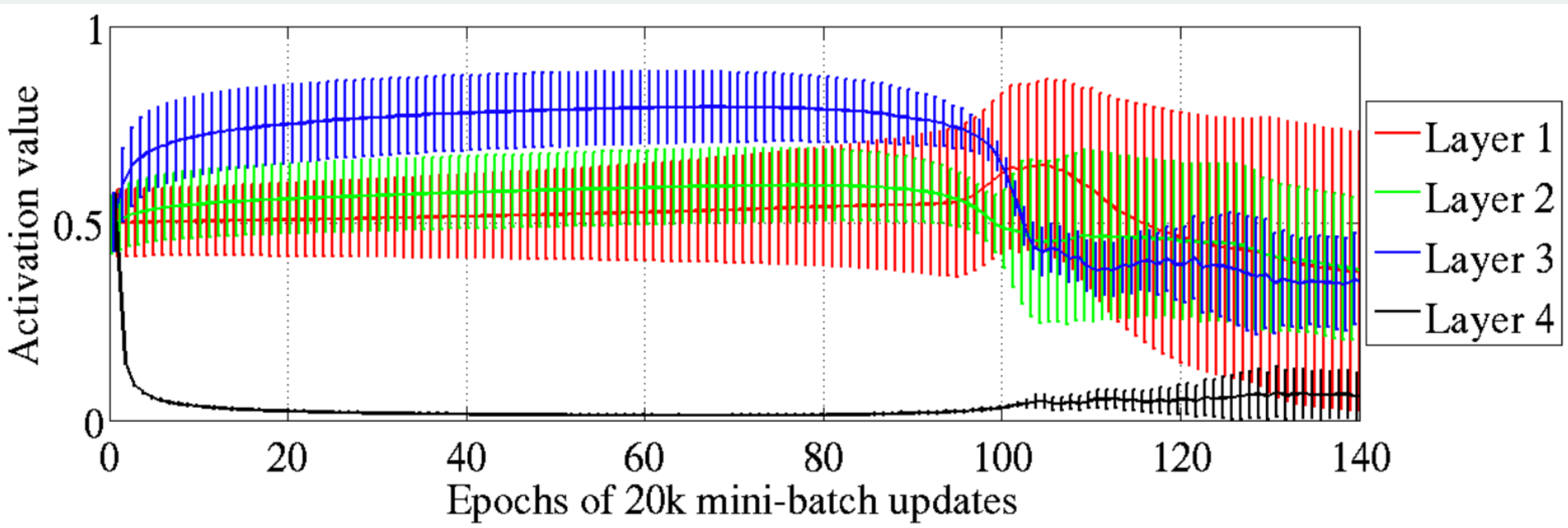
Так как изначально глубокая сеть по сути выдает случайные значения на выходе, простейшим решением будет…



Вывод второй

Tanh лучше Sigmoid.

Почему?

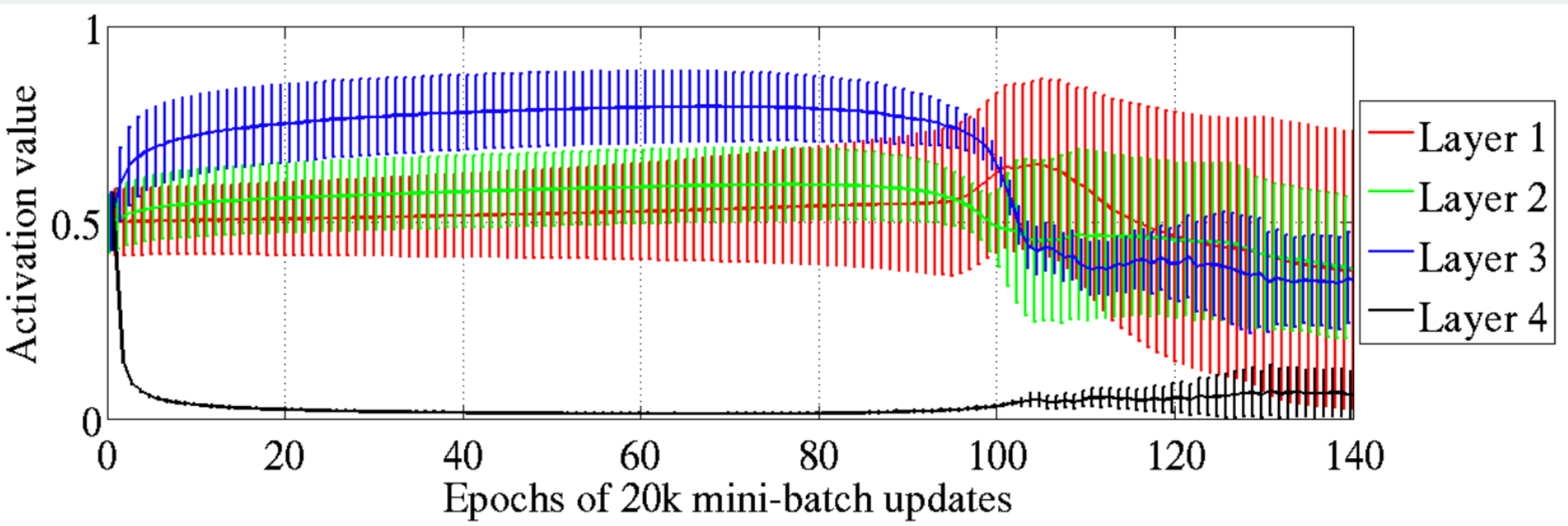


Вывод второй

Tanh лучше Sigmoid.

Когда мы обнуляем выходы последнего слоя мы насыщаем Sigmoid!

А вот Tanh в окрестности нуля чувствует себя отлично.

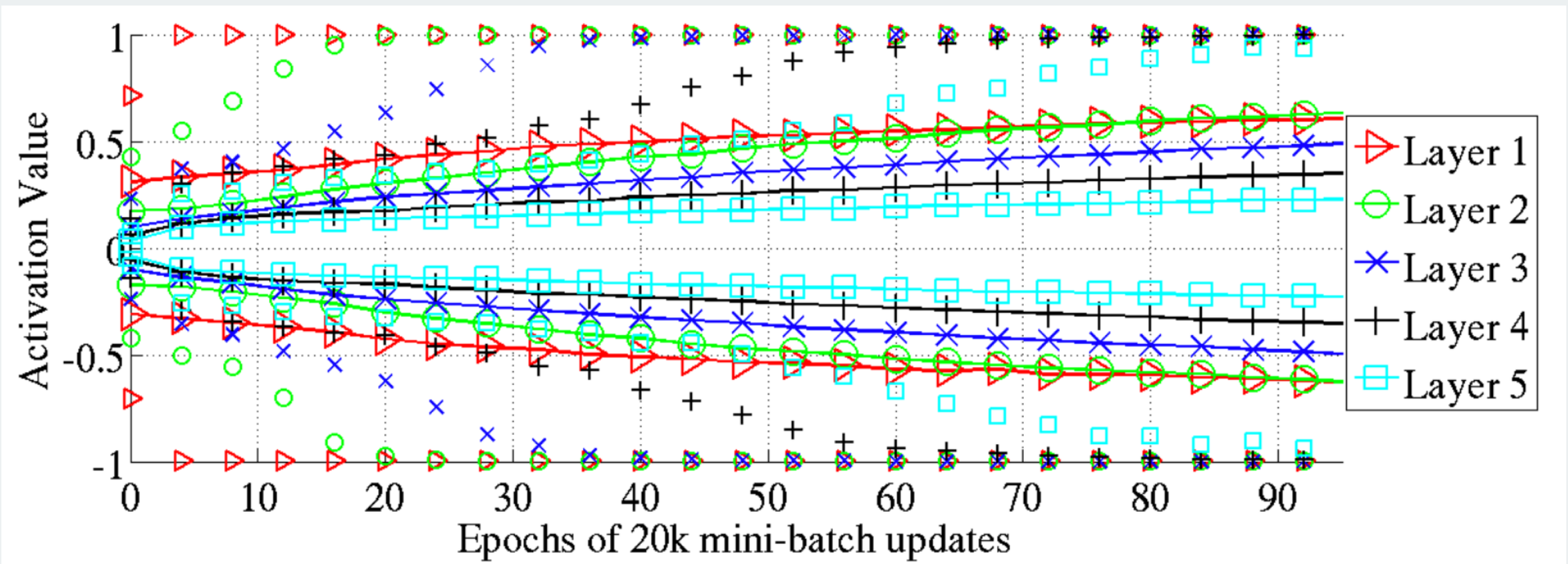


Вывод второй

Tanh лучше Sigmoid.

Когда мы обнуляем выходы последнего слоя мы насыщаем Sigmoid!

А вот Tanh в окрестности нуля чувствует себя отлично.



Но что там с инициализацией?

$$y = w^T x + b = \sum_i w_i x_i + b$$

$Var(x)$ — **известна, тогда** $Var(y) = ?$

Но что там с инициализацией?

$$y = w^T x + b = \sum_i w_i x_i + b$$

$Var(x)$ — **известна, тогда** $Var(y) = ?$

$$Var(y_i) = Var(w_i x_i) = \mathbb{E}\left[w_i^2 x_i^2\right] - (\mathbb{E}[w_i x_i])^2 =$$
$$= \mathbb{E}[x_i]^2 Var(w_i) + \mathbb{E}[w_i]^2 Var(x_i) + Var(w_i)\ Var(x_i)$$

Но что там с инициализацией?

$$y = w^T x + b = \sum_i w_i x_i + b$$

$Var(x)$ — **известна, тогда** $Var(y) = ?$

$$Var(y_i) = Var(w_i x_i) = \mathbb{E}\left[w_i^2 x_i^2\right] - (\mathbb{E}[w_i x_i])^2 =$$
$$= \mathbb{E}[x_i]^2 Var(w_i) + \mathbb{E}[w_i]^2 Var(x_i) + Var(w_i)\ Var(x_i)$$

Если $\mathbb{E}[x_i]^2 = \mathbb{E}[w_i]^2 = 0$**, то** $Var(y_i) = Var(w_i)\ Var(x_i)$

Но что там с инициализацией?

$$y = w^T x + b = \sum_i w_i x_i + b$$

$Var(x)$ — **известна, тогда** $Var(y) = ?$

$$Var(y_i) = Var(w_i x_i) = \mathbb{E}\left[w_i^2 x_i^2\right] - (\mathbb{E}[w_i x_i])^2 =$$
$$= \mathbb{E}[x_i]^2 Var(w_i) + \mathbb{E}[w_i]^2 Var(x_i) + Var(w_i)\ Var(x_i)$$

Если $\mathbb{E}[x_i]^2 = \mathbb{E}[w_i]^2 = 0$**, то** $Var(y_i) = Var(w_i)\ Var(x_i)$

$$Var(y) = Var\left(\sum_{i=1}^{n} y_i\right) = nVar(w_i)\ Var(x_i)$$

Но что там с инициализацией?

$$Var(y) = Var\left(\sum_{i=1}^{n} y_i\right) = nVar(w_i)Var(x_i)$$

$$w_i \sim U\left[-\frac{1}{\sqrt{n_{in}}}, \frac{1}{\sqrt{n_{in}}}\right] \Rightarrow n_{in}Var(w_i) = \frac{1}{3}$$

Если дисперсия активаций от слоя к слою уменьшается до нуля, то чему равны активации последних слоев?

Но что там с инициализацией?

$$Var(y) = Var\left(\sum_{i=1}^{n} y_i\right) = nVar(w_i)Var(x_i)$$

$$w_i \sim U\left[-\frac{1}{\sqrt{n_{in}}}, \frac{1}{\sqrt{n_{in}}}\right] \Rightarrow n_{in}Var(w_i) = \frac{1}{3}$$

Но это не все! При обратном распространении, дисперсия градиентов страдает от той же проблемы! Только вместо количества нейронов на входе будет количество нейронов на выходе.

Предложение

Давайте сохранять уровень дисперсии активаций или градиентов от слоя к слою, то есть, $nVar(w_i)$ должно примерно равняться 1, тогда:

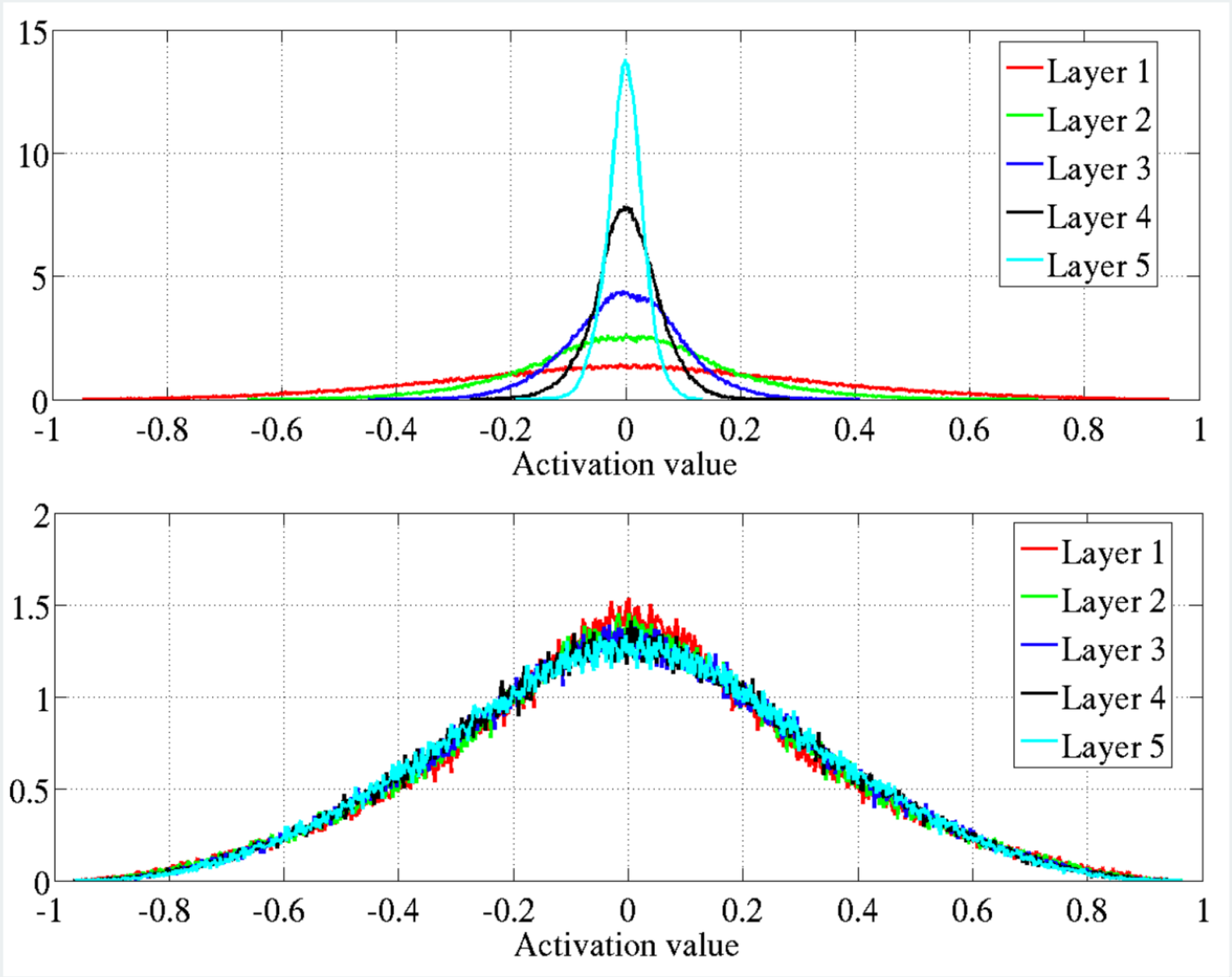
Предложение

Давайте сохранять уровень дисперсии активаций или градиентов от слоя к слою, то есть, $nVar(w_i)$ должно примерно равняться 1, тогда:

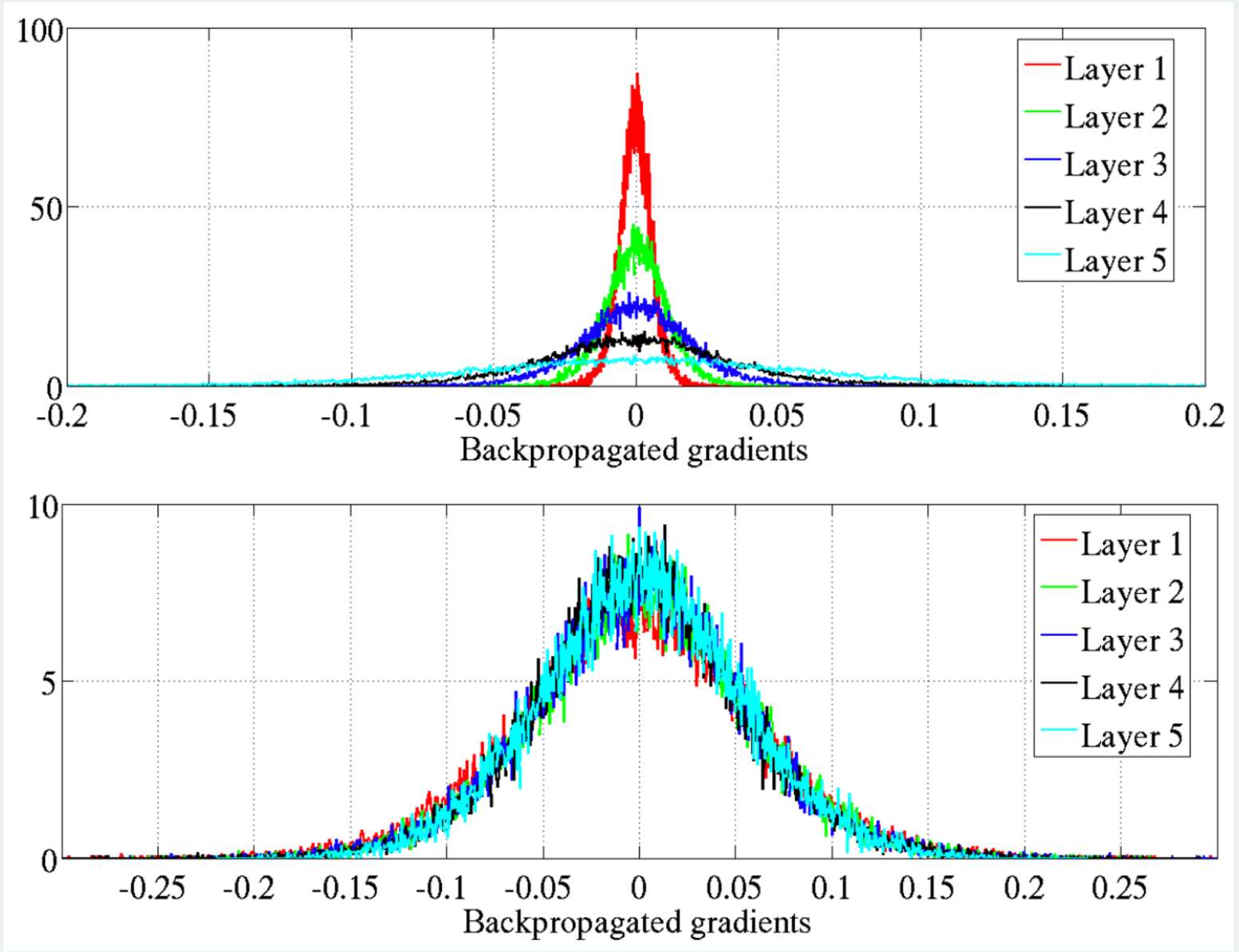
$$w_i \sim U\left[-\frac{\sqrt{6}}{\sqrt{n_{in}+n_{out}}}, \frac{\sqrt{6}}{\sqrt{n_{in}+n_{out}}}\right]$$

$$nVar(w_i) = \frac{2n}{n_{in}+n_{out}} \approx 1$$

После инициализации



После инициализации



Что не так с ReLU?

Так как ReLU не симметрична $\mathbb{E}[x_i] \neq \mathbf{0}$

$$Var(y_i) = Var(w_i x_i) = \mathbb{E}\left[w_i^2 x_i^2\right] - (\mathbb{E}[w_i x_i])^2 =$$
$$= \mathbb{E}[x_i]^2 Var(w_i) + \mathbb{E}[w_i]^2 Var(x_i) + Var(w_i)Var(x_i)$$

Что не так с ReLU?

Так как ReLU не симметрична $\mathbb{E}[x_i] \neq \mathbf{0}$

$$Var(y_i) = Var(w_i x_i) = \mathbb{E}\left[w_i^2 x_i^2\right] - (\mathbb{E}[w_i x_i])^2 =$$
$$= \mathbb{E}[x_i]^2 Var(w_i) + \mathbb{E}[w_i]^2 Var(x_i) + Var(w_i)\, Var(x_i)$$

$$w_i \sim N\left[0, \sqrt{\frac{2}{n_{in}}}\right]$$

K. He et al. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification.

Заключение

1. При обучении глубоких нейросетей важно, чтобы градиенты не затухали и не взрывались.
2. Кроме того, важно, чтобы дисперсия градиентов не затухала.
3. При использовании симметричных функций активации инициализация Ксавье, для ReLU — Хе.

План на сегодня

1. Обратное распространение и градиенты
2. Насыщение функций активации
3. Простые методы борьбы
4. Инициализация весов
5. **Практика**

Спасибо
за внимание!