Глубокое обучение

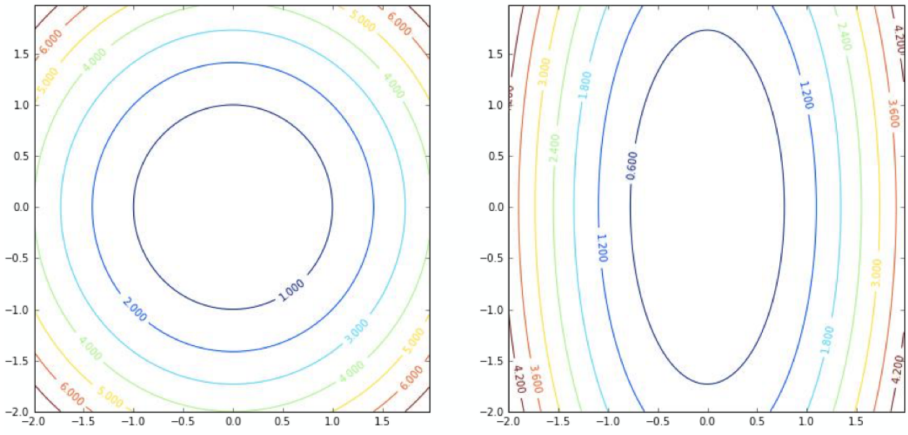
Занятие 3: Batch Normalization & Dropout

Agenda

- Batch Normalization
- Dropout
- Еще эвристики для обучения нейросетей

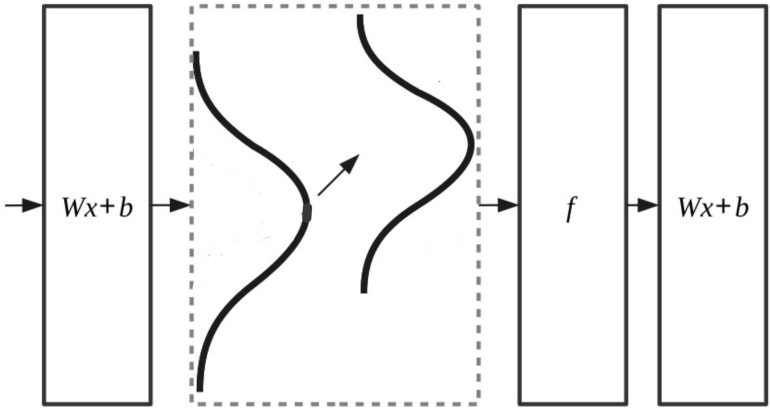
Batch Normalization, она же Батч-нормализация

Стандартизация

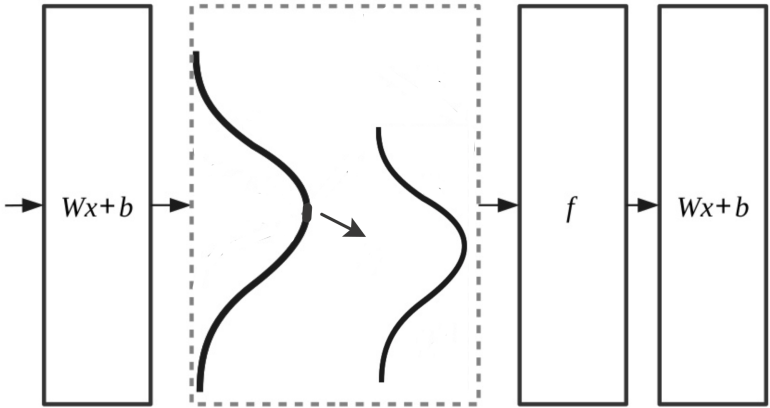


Какая из ситуаций лучше для SGD?

А что внутри?



А что внутри?



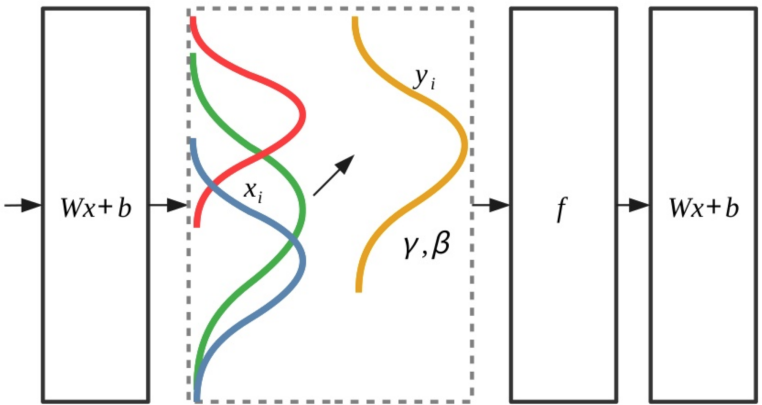
Проблема

- 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 нестандартизован
- Скрытые представления $h = XW + b$ могут менять своё распределение, что осложняет процесс обучения

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 нестандартизован
- Скрытые представления $h = XW + b$ могут менять своё распределение, что осложняет процесс обучения
- 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)



Алгоритм Batch Normalization при обучение

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

Batch Normalization при инференсе

- Откуда взять μ_h и σ_h при инференсе?
- Оценить по каждому из батчей и усреднить определенным образом!
- На этапе валидации используются среднее и дисперсия, посчитанные по всей train выборке

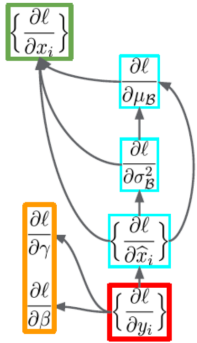
$$\mu_h = \alpha \cdot \mu_{\mathcal{B}} + (1 - \alpha) \cdot \mu_{h-1}$$
$$\sigma_h^2 = \alpha \cdot \sigma_{\mathcal{B}}^2 + (1 - \alpha) \cdot \sigma_{h-1}^2$$

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

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

Backward pass

Стоит обратить вниам ние, что β и γ - НЕ гиперпараметры, а обучаемые веса



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

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

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

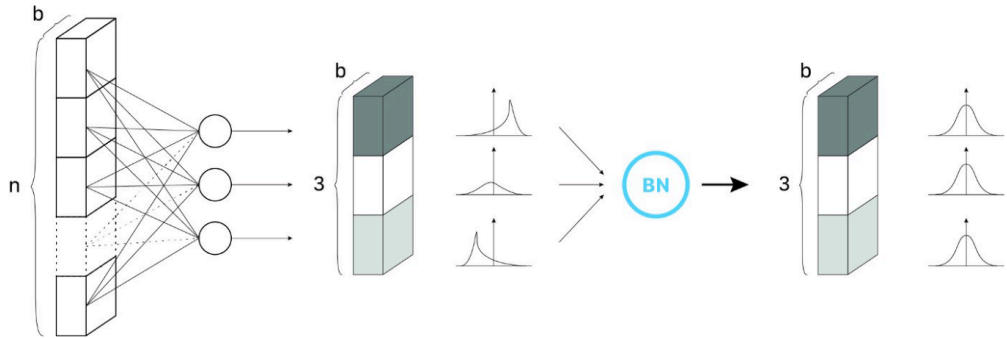
$$\frac{\partial \ell}{\partial x_i} = \frac{\partial \ell}{\partial \widehat{x}_i} \cdot \frac{1}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} + \frac{\partial \ell}{\partial \sigma^2_{\mathcal{B}}} \cdot \frac{2(x_i - \mu_{\mathcal{B}})}{m-1} + \frac{\partial \ell}{\partial \mu_{\mathcal{B}}} \cdot \frac{1}{m}$$

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

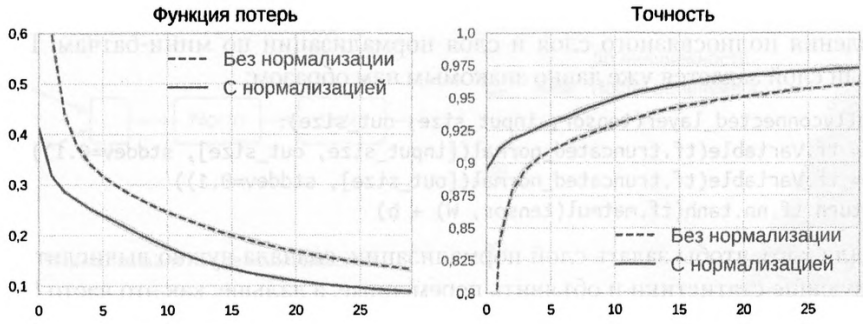
$$\frac{\partial \ell}{\partial \beta} = \sum_{i=1}^{m} \frac{\partial \ell}{\partial y_i}$$

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

Batch norm, еще одна красивая картинка



Эксперимент с MNIST



Источник: Николенко, страница 160

Важные замечания

- С батч-нормализацией нужно уменьшить силу Dropout и регуляризацию
- Не забывайте перемешивать обучающую выборку перед каждой новой эпохой, чтобы батчи были разнообразными

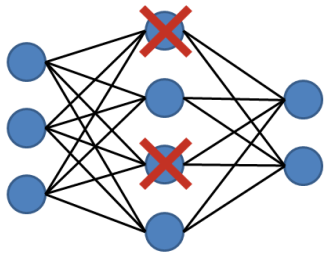
http://openaccess.thecvf.com/content_CVPR_2019/papers/Li_Understanding_the_Disharmony_Between_Dropout_and_Batch_Normalization_by_Variance_CVPR_2019_paper.pdf

Dropout


Overfitting
?!?
Too many neurons
Not enough DATA
BAD Network

Dropout

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



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

Dropout в формулах

- forward pass:

$$o = f(X \cdot W + b)$$

$*$ - elementwise multiplication, т.е. поэлементное умножение

Dropout в формулах

- forward pass:

$$o = f(X \cdot W + b)$$
$$o = D * f(X \cdot W + b), \quad D = (D_1, \dots, D_k) \sim iidBern(p)$$

$*$ - elementwise multiplication, т.е. поэлементное умножение

Dropout в формулах

- forward pass:

$$o = f(X \cdot W + b)$$
$$o = D * f(X \cdot W + b), \quad D = (D_1, \dots, D_k) \sim iidBern(p)$$

$$o_i = D_i \cdot f(wx_i^T + b) = \begin{cases} f(wx_i^T + b), p \\ 0, 1 - p \end{cases}$$

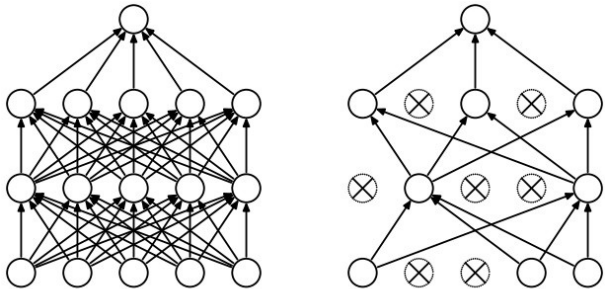
Дропаут — это просто небольшая модификация функции активации

$*$ - elementwise multiplication, т.е. поэлементное умножение

Смысл Dropout: усреднение работы нескольких сетей

- 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.

Смысл Dropout: усреднение работы нескольких сетей

- 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 инференс?

Смысл Dropout: усреднение работы нескольких сетей

- 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

Смысл Dropout: усреднение работы нескольких сетей

- 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

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

Inverted Dropout

- При инференсе ищем математическое ожидание:

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

Inverted Dropout

- При инференсе ищем математическое ожидание:

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

- Это неудобно! Надо переписывать функцию для прогнозов!

Inverted Dropout

- При инференсе ищем математическое ожидание:

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

- Это неудобно! Надо переписывать функцию для прогнозов!

- Давайте лучше будем домножать на $\frac{1}{p}$ на этапе обучения (техника Inverted Dropout):

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

Inverted Dropout:
https://www.coursera.org/lecture/deep-neural-network/dropout-regularization-eM33A
$*$ - elementwise multiplication, т.е. поэлементное умножение

Почему Inverted Dropout работает?

- Рассмотрим для примера 4-ый слой в нейросети

$$z^{[4]} = w^{[4]} \cdot a^{[3]} + b^{[4]}$$

- $a^{[3]}$ - это выход с 3-го слоя

$$a^{[3]} = f(w^{[3]} \cdot a^{[2]} + b^{[3]}) * d^{[3]}, \quad d^{[3]} = \begin{cases} 1, p \\ 0, 1-p \end{cases}$$

- пусть $p = 0.8$, тогда 20% нейронов занулится, т.е. $\mathbb{E}\{a^{[3]}\}$ уменьшится на 20%

- чтобы этого избежать будем делить $a^{[3]}$ на p на этапе обучения

$*$ - elementwise multiplication, т.е. поэлементное умножение

Другие эвристики для обучения сеток

Предобучение

- На будущее: обучаем на корпусе картинок автокодировщик, encoder благодаря этому учится выделять наиболее важные фичи, которые позволяют эффективно сжимать изображения. После срезаем decoder и на его месте достраиваем слои для решения нашей задаче, запускаем обычное дообучение.

Динамическое наращивание сети

- Обучение сети при заведомо недостаточном числе нейронов H
- 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
- 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.

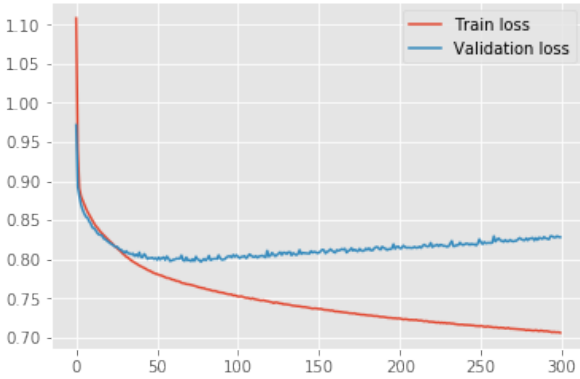
Прореживание сети

- Начать с большого количество нейронов и удалять незначимые по какому-нибудь критерию
- Пример: обнуляем вес, смотрим как сильно упала ошибка, сортируем все связи по этому критерию, удаляем N наименее значимых
- После прореживания снова запускаем backprop
- Если качество модели сильно упала, надо вернуть последние удалённые связи

Более понятные идеи:

- Ранняя остановка процесса обучения
- l_1 и l_2 регуляризация
- Различные новые градиентные спуски, ускоряющие процедуру сходимости

Early stopping



- Будем останавливать обучение, когда качество на валидации начинает падать

Регуляризация

- L_2: приплюсовываем к функции потерь $\lambda \cdot \sum w_i^2$
- L_1: приплюсовываем к функции потерь $\lambda \cdot \sum |w_i|$
- Можно регуляризовать не всю сетку, а отдельный нейрон или слой
- Не даёт нейрону сфокусироваться на слишком выделяющемся входе

Взаимное использование изученных техник

- На практике для регуляризации обычно используют Dropout.
- Например, в [1] написано:

  «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»

- У Гудфеллоу в Глубоком обучении на стр. 218 можно найти, что ранняя остановка для линейных моделей эквивалентна l_2 регуляризации с MSE, обучаемой SGD.

[1] https://arxiv.org/abs/1307.1493

Что еще помогает при обучении сетей

В дальнейшем пройдем:

- Скип-конекшены
- Аугментация данных
- Более изощренные архитектуры

Семинар с практикой по BatchNorm & Dropout