Машинное обучение (Machine Learning)

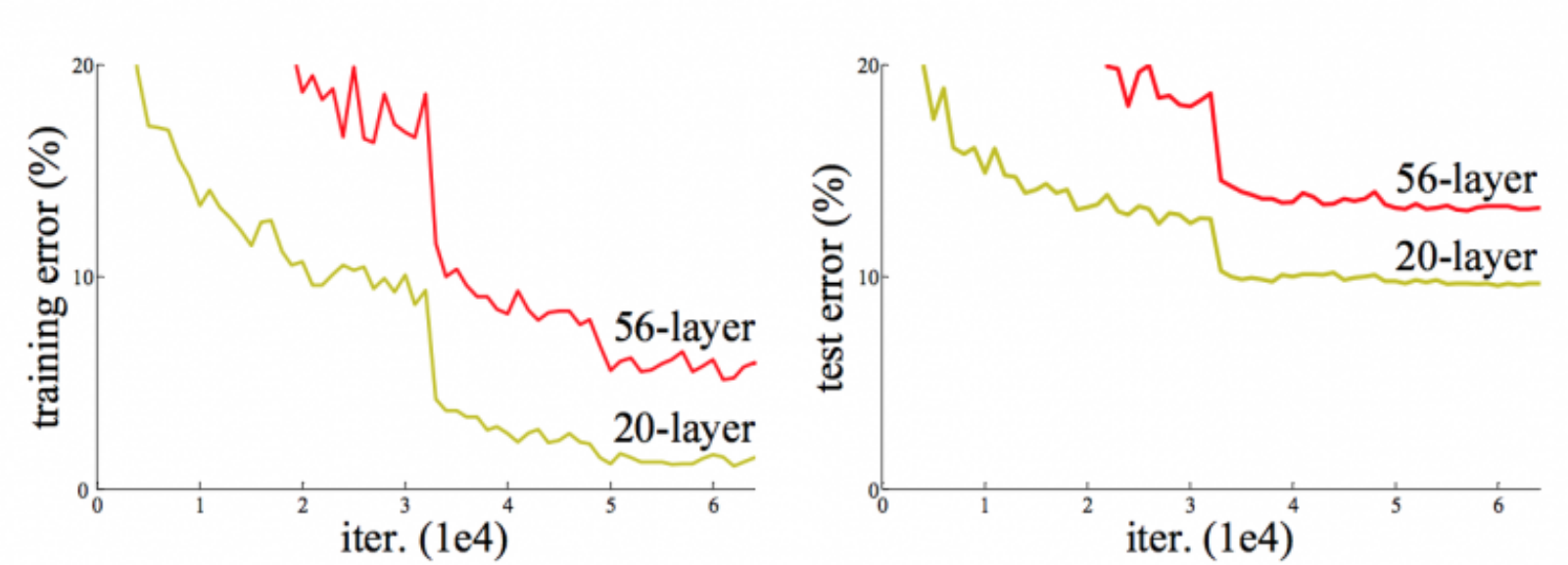
Архитектуры глубоких нейронных сетей

Уткин Л.В.

Санкт-Петербургский политехнический университет Петра Великого

ResNet (1)

- ResNet - https://arxiv.org/pdf/1512.03385.pdf



Ошибка обучения (слева) и ошибка теста (справа) на CIFAR-10 с 20-уровневыми и 56-слойными «простыми» сетями

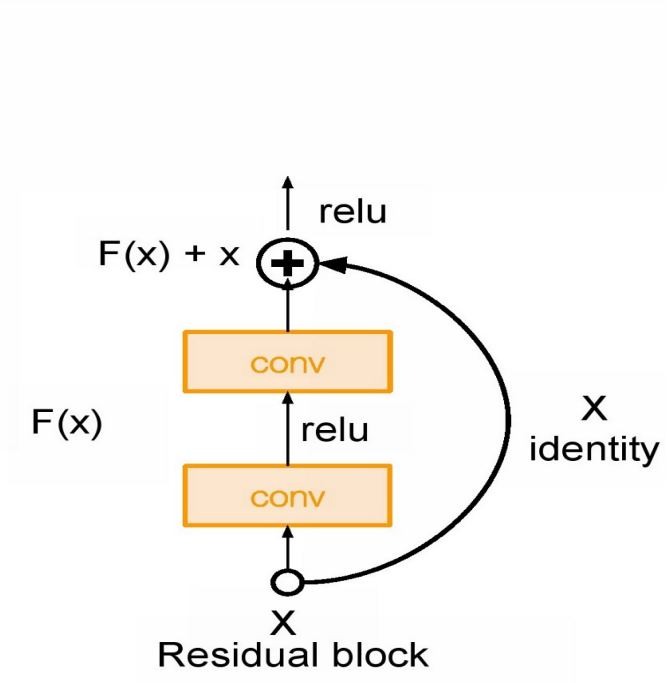
ResNet (2)

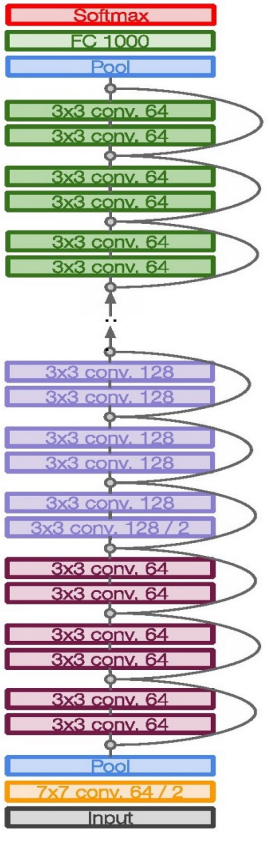
- С увеличением глубины сети точность сначала увеличивается, а затем быстро ухудшается.
- Почему глубокие сети так себя ведут?
  - Исчезающий градиент
  - Взрывной градиент
  - Проблема деградации

ResNet (3)

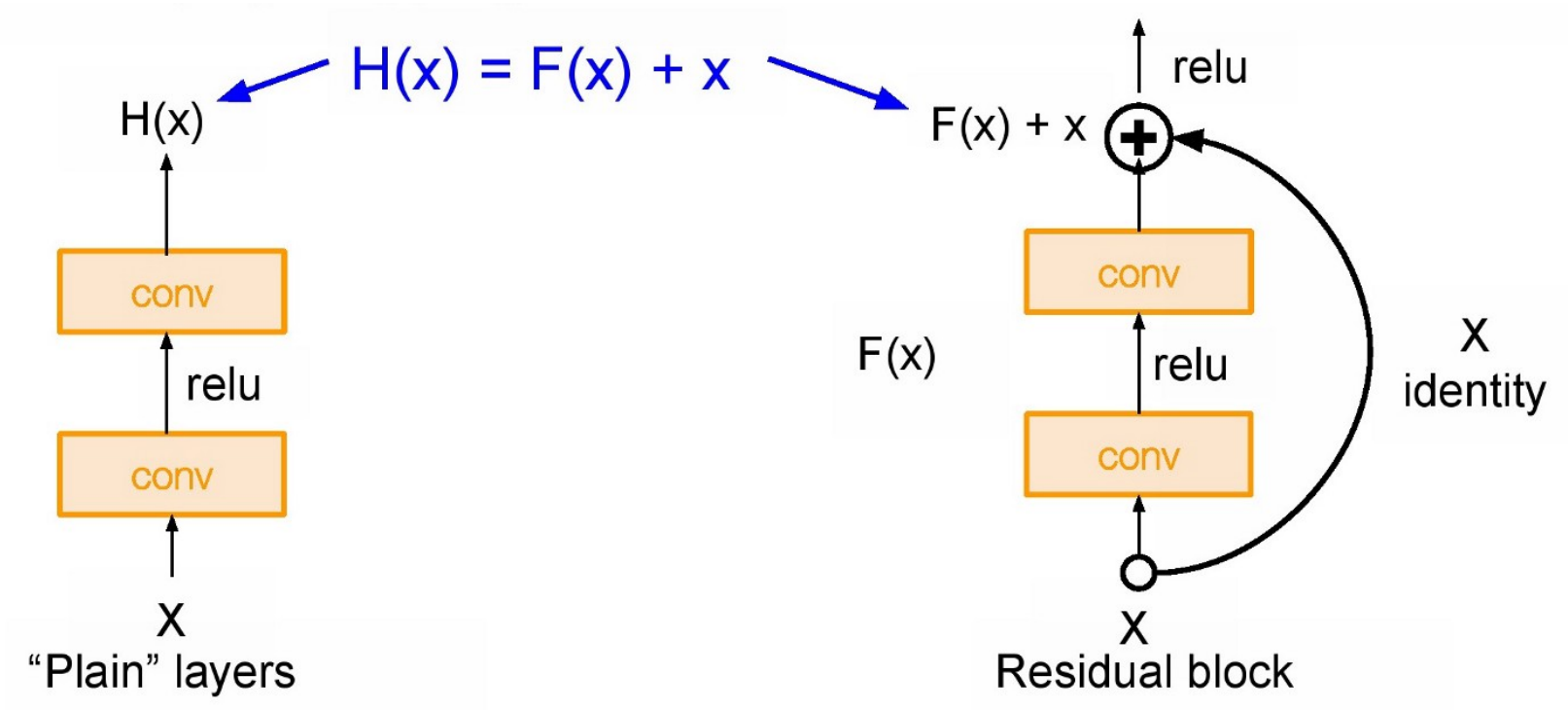
- Решение проблемы деградации - **соединение для быстрого доступа**
- Соединения быстрого доступа (**identity mapping, skip connections, shortcut connections**) пропускают один или несколько слоев и выполняют сопоставление идентификаторов. Их выходы добавляются к выходам stacked layers.

ResNet (shortcut connections)

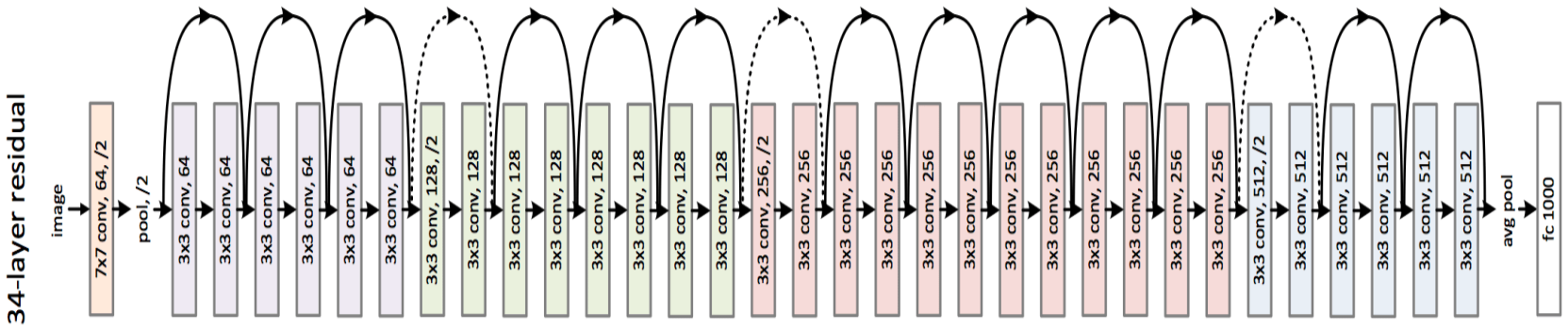




ResNet (shortcut connections)



ResNet (архитектура всей сети)



В качестве финального классификатора в ResNet используется pooling-слой с softmax.

ResNet (проблема разной размерности)

- В идеале размерность $F(\mathbf{x})$ и $\mathbf{x}$ должны быть одинаковыми, но ...
- Вопрос - как можно добавить $F(\mathbf{x})$ и $\mathbf{x}$, когда два слоя имеют разные размеры?
- Когда размерности увеличиваются - с помощью заполнения нулями или линейной проекции на соединениях быстрого доступа.
- В ResNet существует два вида быстрых соединений: одно без весов, а другое с весами.

ResNet (проблема разной размерности - решение)

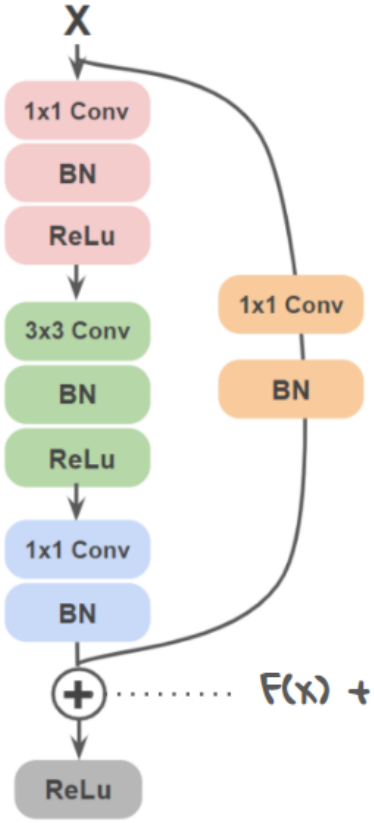
- Быстрое соединение выполняет сопоставление идентификаторов с дополнительными нулями, добавленными для увеличения размерности. Эта опция не вводит никаких дополнительных параметров
- Проекция быстрого соединения в $F(\mathbf{x}, \{W_i\}) + W_s\mathbf{x}$ используется для сопоставления размерностей (выполнено с помощью 1x1 сверток)

ResNet (проекция быстрого соединения)

- Формально один блок имеет функцию:
  $\mathbf{y} = F(\mathbf{x}, \{W_i\}) + \mathbf{x}$, где $\mathbf{x}$, $\mathbf{y}$ - вход и выход блока;
  $F(\mathbf{x}, \{W_i\})$ - функция, например, $F = W_2\sigma(W_1\mathbf{x})$, σ - ReLU.
- Операция $F + \mathbf{x}$ реализуется быстрым соединением и поэлементным суммированием
- Линейная проекция W_s: $\mathbf{y} = F(\mathbf{x}, \{W_i\}) + W_s\mathbf{x}$

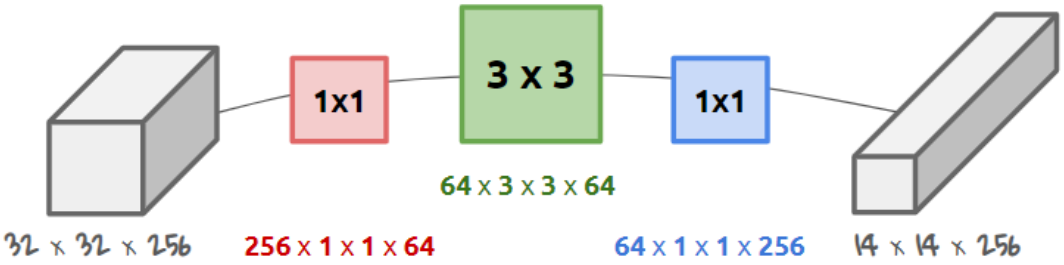
ResNet (проекция быстрого соединения)

< A bottleneck building for ResNet-50/101/152 >



F(x) + WsX

=> projection shortcuts



- (32 x 32 x 256) * (256 x 1 x 1 x 64) -> 32 x 32 x 64
- (32 x 32 x 64) * (64 x 3 x 3 x 64) -> 14 x 14 x 64
- (14 x 14 x 64) * (64 x 1 x 1 x 256) -> 14 x 14 x 256

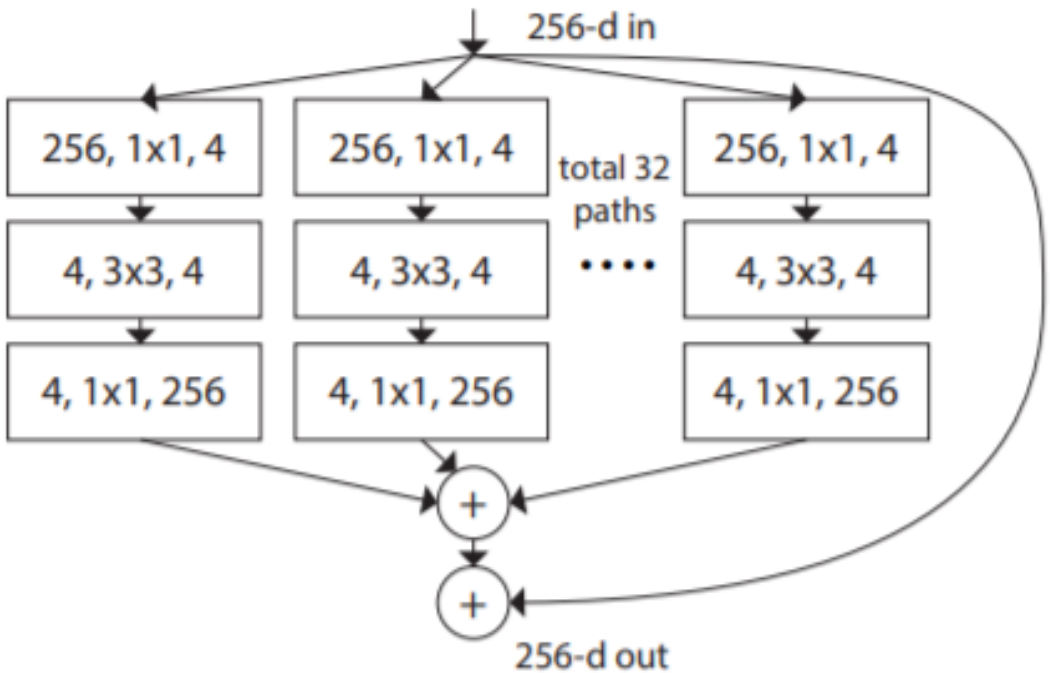
Reducing

Increasing

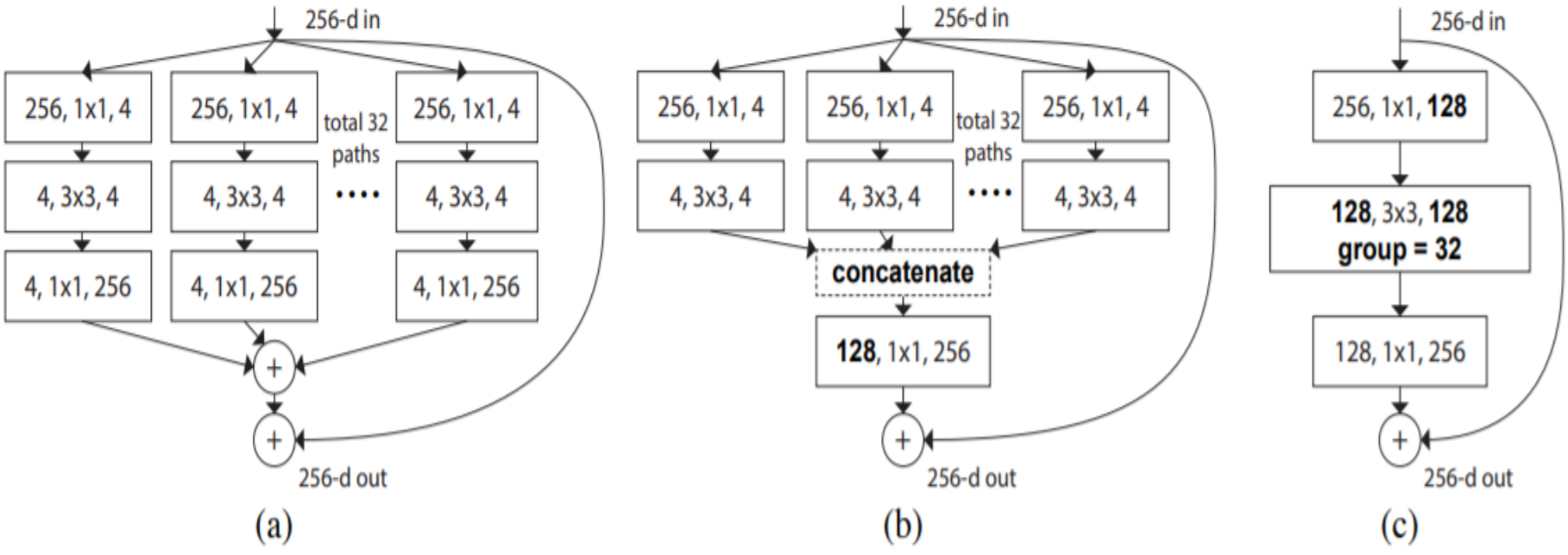
" Bottleneck "

ResNeXt

Использует стратегию разделения-преобразования-слияния. Блок выглядит как Inception, где выполняются различные преобразования (1x1 Conv, 3x3 Conv, 5x5 Conv, MaxPooling) и объединяются вместе

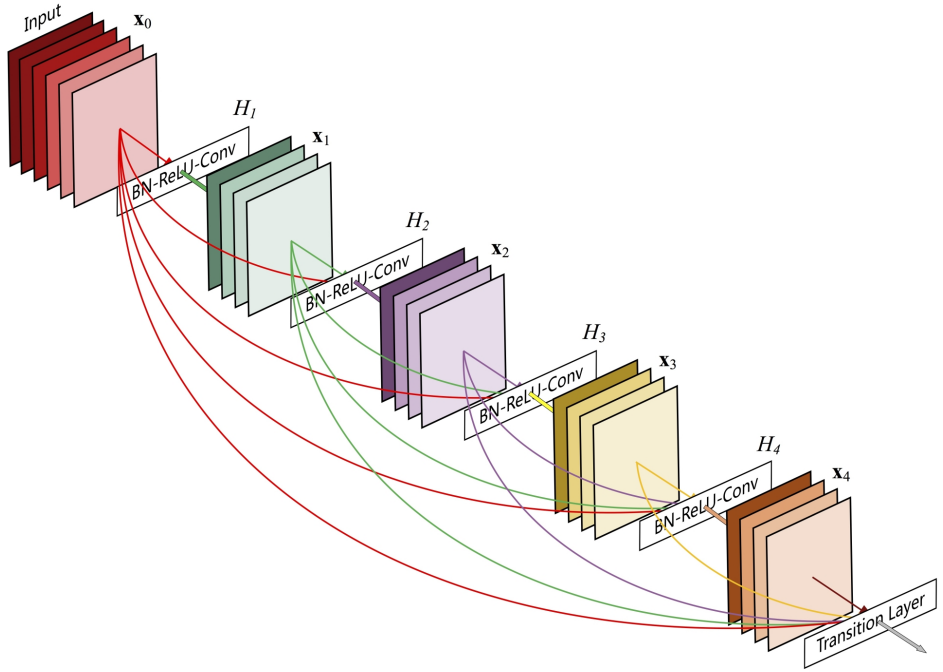


ResNeXt (еще варианты)



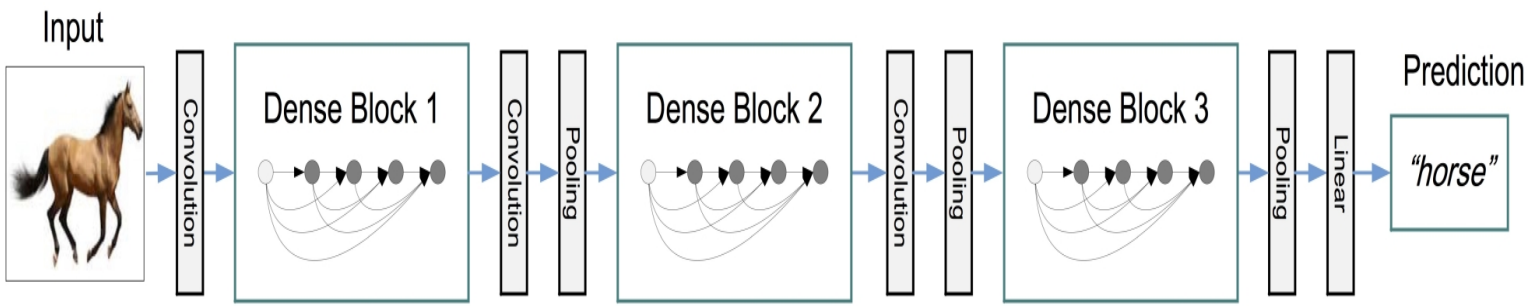
DenseNet (один dense блок)

- Один плотный блок DenseNet с 5 слоями и скоростью роста $k = 4$
- Каждый слой принимает все предыдущие карты признаков в качестве входных данных.



DenseNet

DenseNet с тремя плотными блоками



В отличие от ResNet, признаки не суммируются, а конкатенируются.

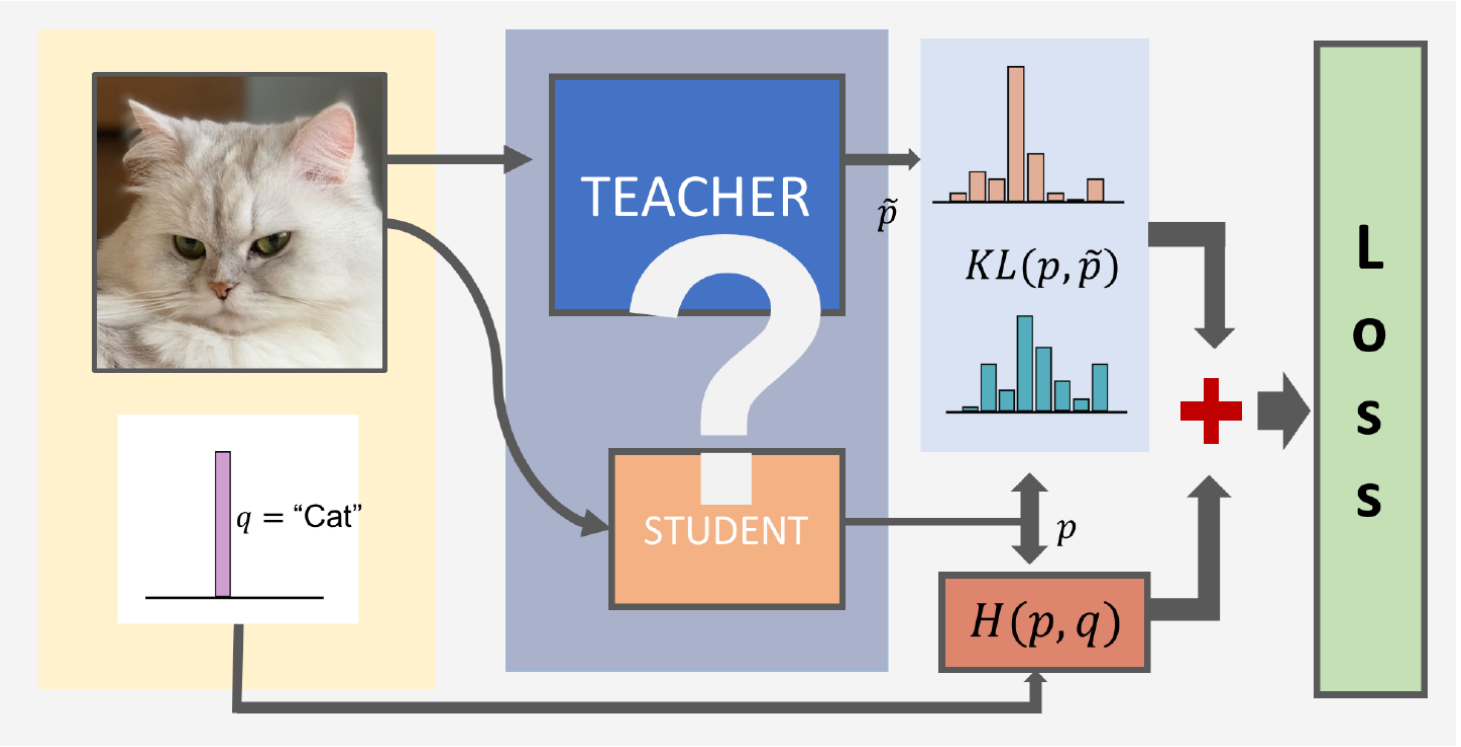
Distillation (idea)

- G. Hinton, O. Vinyals, J. Dean. Distilling the Knowledge in a Neural Network, arXiv:1503.02531.
- Мотивация: использовать результаты обучения "сложной" модели для обучения "простой"
- Отличие transfer learning и distillation: в последнем перенос обобщения (transfer of generalization)
- Понятия: сети учитель и студент, понятие температуры в softmax, "темные" знания (dark knowledge) и мягкие вероятности

Distillation (модель учитель-студент)

- Учитель - это “сложная” глубокая нейронная сеть, которая была обучена на большом количестве данных (или любая другая модель - ансамбль) с хорошим обобщением
- Студент - это “простой” сеть, цель - выучить большинство обобщений учителя и оставаться “простой”

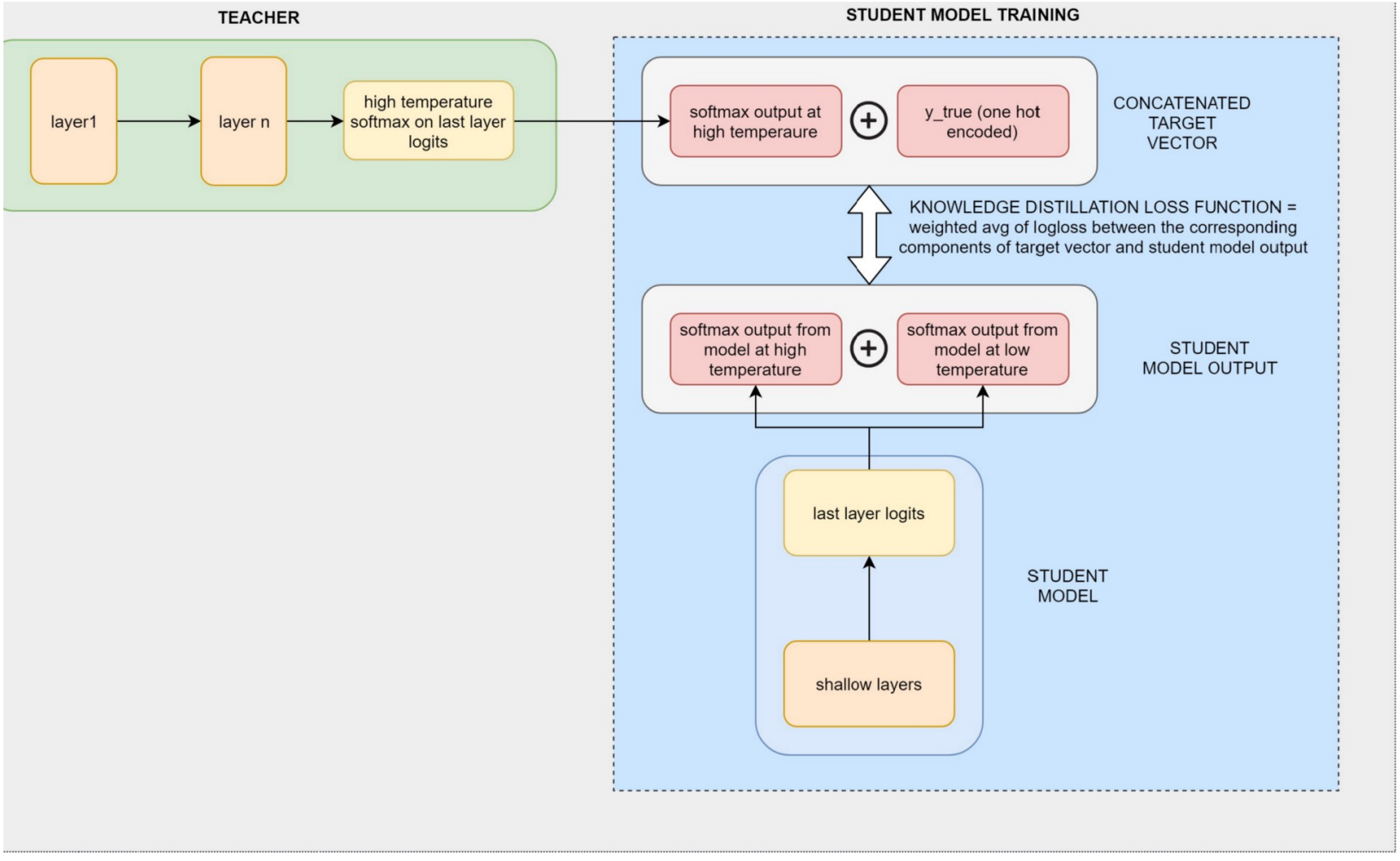
Distillation (модель учитель-студент)



J.H. Cho and B. Hariharan. On the Efficacy of Knowledge Distillation.

arXiv:1910.01348v1

Distillation (модель учитель-студент)



Distillation (температура)

- Softmax возвращает вероятности каждого класса от 0 до 1, и их сумма =1, целевой класс имеет высокую вероятность

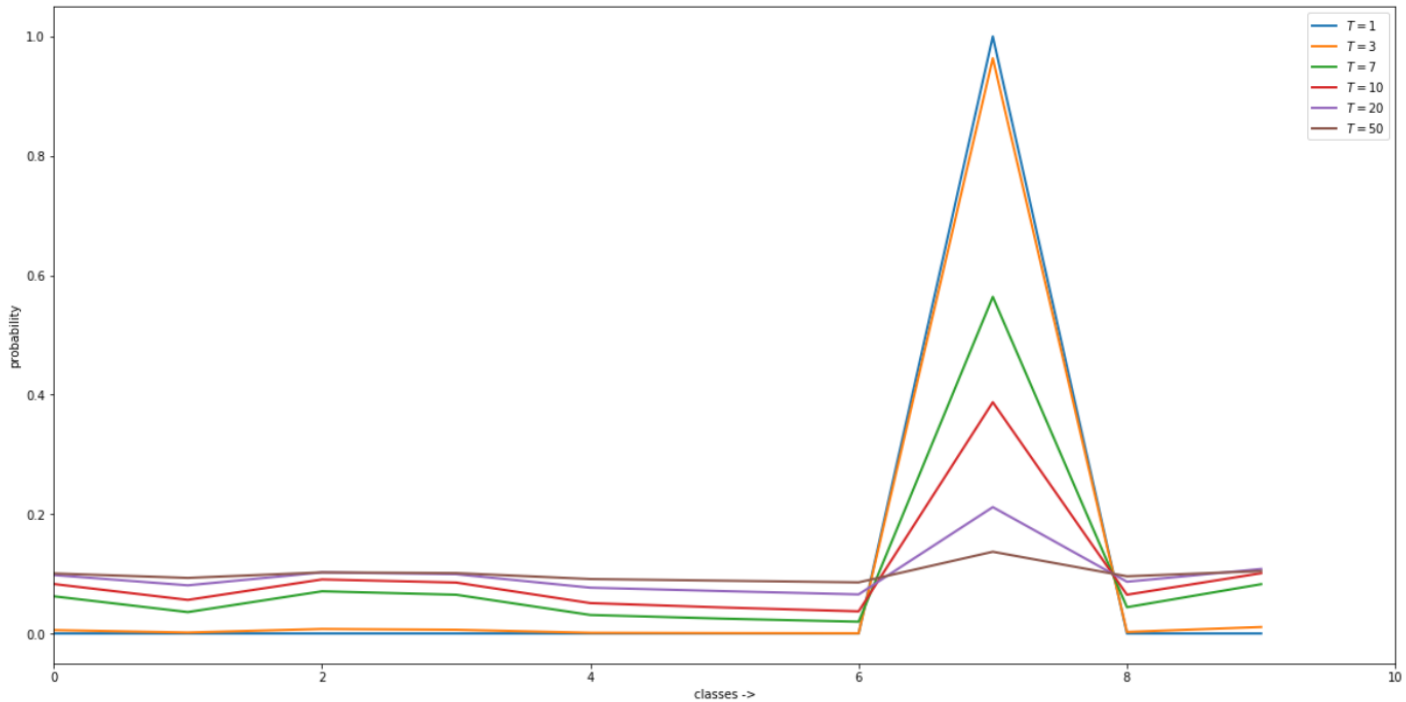
$$p_t(z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

- Softmax с температурой

$$p_t^*(z_i, T) = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$$

- Больше температура - более размыты вероятности классов

Distillation (температура и MNIST)



Вероятности цифр MNIST, классифицируется цифра 7

Distillation (dark knowledge)

один семь или один?

Distillation (dark knowledge)

- Модель дает более высокую вероятность для 1 одновременно прогнозируя 7 при высокой T
- Человек не может количественно определить, насколько 7 выглядит ближе к 1, а “высокотемпературная” модель делает это
- Т.о. “высокотемпературная” модель обладает “темными” знаниями - в дополнение к предсказанию числа 7, она также хранит информацию о том, насколько это число 7 напоминает число 1
- “Низкотемпературная” модель (обычная модель) хороша для точных прогнозов, но теряем эти “темные” знания
- Основная идея distillation - передача “темных” знаний от обученного учителя к простой модели студента

Distillation (обучение студента)

- Модель студента обучается при той же высокой температуре, что и учитель
- Функция потерь для студента

$$L = \alpha L_{\text{cross entopy}} + (1 - \alpha) L_{\text{knowledge distil.}}$$

$$L_{\text{knowledge distil.}} = -\tau \sum_i p_t^*(z_i, T) \ln p_s^*(z_i, T)$$

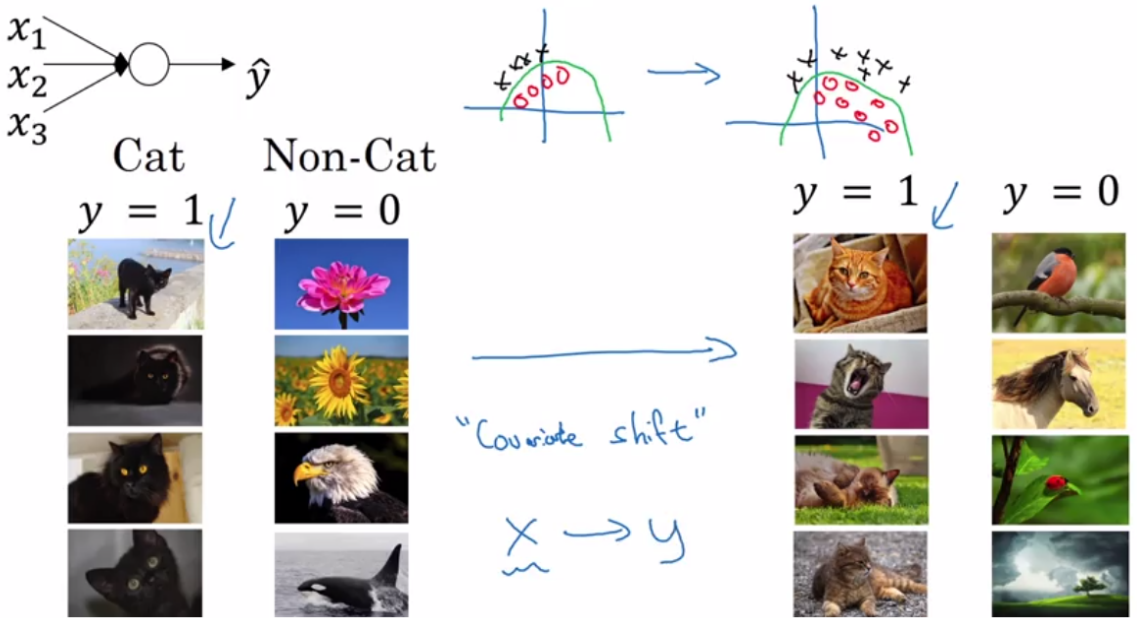
- Модель студента тестируется с обычной активацией softmax (т.е. без температуры).

Batch normalization (пакетная нормализация, BN) - зачем

- Есть функции активации от 0 до 1, а есть от 1 до 1000
- Если нормализуем входной слой, почему не сделать это для всех или части слоев
  - Это добавляет некторый шум к активациям аналогично dropout (регуляризация)
  - Уменьшает смещение
  - Делает слои сети более независимыми от других слоев
  - Более высокая скорость обучения, т.к. не будет очень больших или малых активации

BN - уменьшает смещение

Сеть по классификации кошек: обучаем только на черных кошках. Если применить сеть к цветными кошками, то будут ошибки. Обучающий и тестовый датасеты немного различаются. Batch normalization уменьшает смещение



Deeplearning.ai: Why Does Batch Norm Work? (C2W3L06)

BN - как

BN добавляет два обучаемых параметра к каждому слою: γ и β, что позволяет SGD выполнять денормализацию, изменяя только эти два веса для каждой активации, вместо потери стабильности сети путем изменения всех весов

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

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

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

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

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

S. Ioffe, C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift // arXiv:1502.03167v3

BN - зачем gamma и beta

Если использовать BN в предобученной сети, то это изменит обученные веса (плохо)
Поэтому нужно определить γ и β, чтобы отменить изменение выходных данных

Вопросы

?