

Сверточные сети большой глубины для распознавания изображений

Диц Даниил Денисович

НТУ Сириус

24 марта 2025 г.

Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition //arXiv preprint arXiv:1409.1556. – 2014.

- Разработана в 2014 году Карен Симонян и Эндрю Зиссерманом
- Названа в честь Группы визуальной геометрии (VGG) Оксфордского университета
- Значительное улучшение по сравнению с предыдущими моделями в ILSVRC 2014
- Ключевые особенности:
 - Продемонстрировала преимущества очень глубоких сетей
 - Установила простые, единообразные архитектурные принципы
 - Достигла ошибки top-5 7.3% на ImageNet

Архитектура сети VGG

Основные характеристики

- Вход: 224×224 RGB изображения
- Свертка: ядра 3×3
- Макс-пулинг: окна 2×2
- Прогрессия каналов:
 - $64 \rightarrow 128 \rightarrow 256 \rightarrow 512 \rightarrow 512$

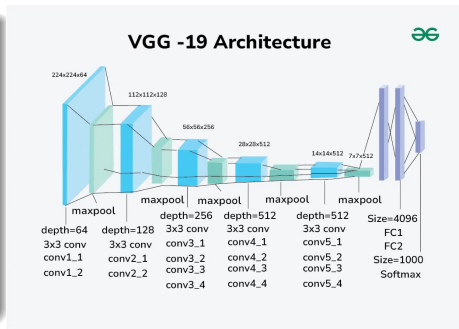


Рис.: Структура сети VGG

Варианты VGG

Тип слоя	VGG-16	VGG-19
Сверточные	13	16
Полносвязные	3	3
Всего слоев	16	19
Параметры	138M	144M

Таблица: Сравнение архитектур VGG

- Все конфигурации используют:
 - Активацию ReLU
 - Макс-пулинг 2×2
 - 3 полносвязных слоя

Важность глубины

- Гораздо глубже предыдущих моделей (AlexNet: 8 слоев)
- Продемонстрировала преимущества увеличения глубины с малыми фильтрами

Малые рецептивные поля

- Стек сверток 3×3 вместо больших фильтров
- Преимущества:
 - Меньше параметров
 - Больше нелинейностей (ReLU)
 - Лучшее обучение признакам

- **Мультимасштабное обучение:**

- Случайное масштабирование изображений
- Улучшает инвариантность к масштабу

- **Аугментация данных:**

- Горизонтальное отражение
- Случайное кадрирование
- Изменение RGB

- **Регуляризация:**

- Dropout ($p=0.5$) в FC слоях
- L2 регуляризация

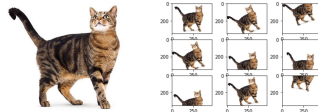


Рис.: Аугментация

Пример реализации на PyTorch

Сравнение производительности

Модель	Ошибка Тор-1	Ошибка Тор-5
AlexNet (2012)	37.5%	17.0%
VGG-16 (2014)	28.5%	9.9%
VGG-19 (2014)	28.7%	9.9%

Таблица: Ошибки классификации на ImageNet

Ключевые преимущества

- Проще архитектура, чем у конкурентов
- Лучшая обобщающая способность
- Легче модифицировать и расширять

- **Повлияла** на последующие архитектуры (ResNet и др.)
- **Широко используется** как экстрактор признаков
- **Доказала** эффективность глубоких сетей
- **Установила** стандарт сверток 3×3

Ограничения

- Высокие вычислительные затраты
- Большие требования к памяти
- Позже превзойдена остаточными сетями

- VGG продемонстрировала важность глубины в CNN
- Простая, единообразная архитектура достигла state-of-the-art результатов
- Установленные принципы проектирования актуальны до сих пор
- Основа для современных архитектур глубокого обучения

CosineAnnealingLR: адаптивный learning rate

Основная идея

Learning rate изменяется по косинусоидальному закону:

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min}) \left(1 + \cos \left(\frac{T_{cur}}{T_{max}} \pi \right) \right)$$

- η_{max} - начальный LR
- η_{min} - минимальный LR
- T_{cur} - текущая эпоха
- T_{max} - общее число эпох

Реализация в PyTorch

```
from torch.optim.lr_scheduler import CosineAnnealingLR

optimizer = torch.optim.SGD(
    model.parameters(), lr=0.1)
scheduler = CosineAnnealingLR(
    optimizer, T_max=100, eta_min=0.001)

for epoch in range(100):
    train(...)
    scheduler.step()
```