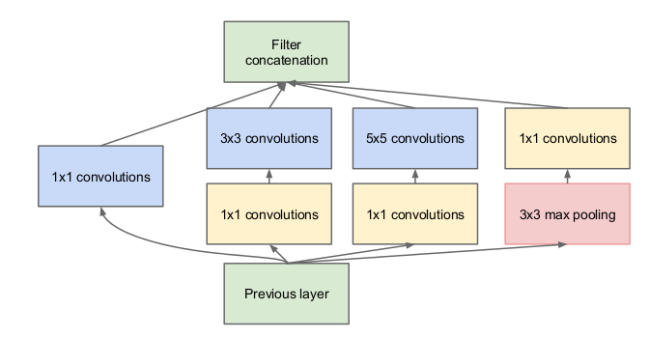
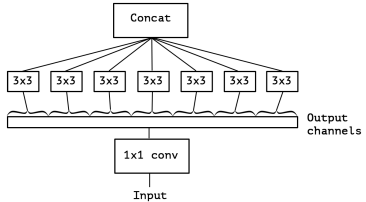
**Текст для призентации**

**Модуль Inception**  
В 2015 году была предложена архитектура Inception, идея которой заключалась в следующем: давайте вместо того, чтобы выбирать размер ядра, возьмём несколько вариантов сразу, используем их все одновременно и конкатенируем результаты. Однако это существенно увеличивает количество операций, которые необходимо выполнить для вычисления активаций одного слоя, поэтому авторы [оригинальной статьи](https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Szegedy_Going_Deeper_With_2015_CVPR_paper.pdf) предлагают такую хитрость: давайте перед каждым свёрточным блоком делать свёртку с размером ядра 1х1, снижая размерность сигнала, подающегося на вход свёрткам с б**о**льшими размерами ядер.  
  
Получившаяся на рисунке снизу конструкция и составляет полный модуль Inception.



**Depthwise separable convolution**  
Представим, что мы взяли стандартный свёрточный слой с фильтрами размера 3х3, на вход которому подается тензор размерности , где — это ширина и высота тензора, а — количество каналов.   
Что делает такой слой? Он сворачивает одновременно все каналы исходного сигнала разными свёртками. На выходе у такого слоя получается тензор размерности .  
  
Давайте вместо этого сделаем последовательно два шага:  


1. Свернём исходный тензор 1х1 свёрткой, подобно тому как мы делали в блоке Inception, получив тензор
2. . Эта операция называется pointwise convolution
3. Свернём каждый канал по отдельности 3х3 свёрткой (при этом размерность не изменится, так как мы сворачиваем не все каналы вместе, как в обычном свёрточном слое). Эта операция называется depthwise spatial convolution

**Немного занудства о терминологии.**

**Почему это делает сеть компактнее?**

Давайте разберём конкретный пример. Пусть мы сворачиваем изображение с 16 каналами свёрточным слоем с 32 фильтрами. Суммарно этот свёрточный слой будет иметь весов, так как у нас будет свёрток 3х3.  
  
Сколько же весов будет в аналогичном depthwise separable convolution блоке? Во-первых, у нас будет весов у pointwise convolution. Во-вторых, у нас будет весов у depthwise convolution. В сумме получим 800 весов, что намного меньше, чем у обычного свёрточного слоя.

**Почему это вообще работает?**

Обычный свёрточный слой одновременно обрабатывает как пространственную информацию (корреляцию соседних точек внутри одного канала), так и межканальную информацию, так как свёртка применяется ко всем каналам сразу. Архитектура Xception базируется на предположении о том, что эти два вида информации можно обрабатывать последовательно без потери качества работы сети, и раскладывает обычную свёртку на pointwise convolution (которая обрабатывает только межканальную корреляцию) и spatial convolution (которая обрабатывает только пространственную корреляцию в рамках отдельного канала).  
  
Посмотрим на реальный эффект. Для сравнения возьмём две по-настоящему глубоких архитектуры свёрточных сетей — ResNet50 и InceptionResNetV2.   
  
ResNet50 имеет 25 636 712 весов, а предобученная модель в Keras весит 99 Мб. Точность, которая достигается этой моделью на датасете ImageNet, составляет 75.9%.  
  
InceptionResNetV2 имеет 55 873 736 обучаемых параметров и весит 215 Мб, достигая точности 80.4%.  
  
Что же получается с архитектурой Xception? Сеть имеет 22 910 480 весов и весит 88 Мб. При этом точность классификации на ImageNet составляет 79%.  
  
Таким образом, мы получаем архитектуру сети, которая превосходит по точности ResNet50 и лишь чуть-чуть уступает InceptionResNetV2, при этом **существенно выигрывая по размерам**, а значит по требуемым ресурсам как для обучения, так и для использования этой модели.