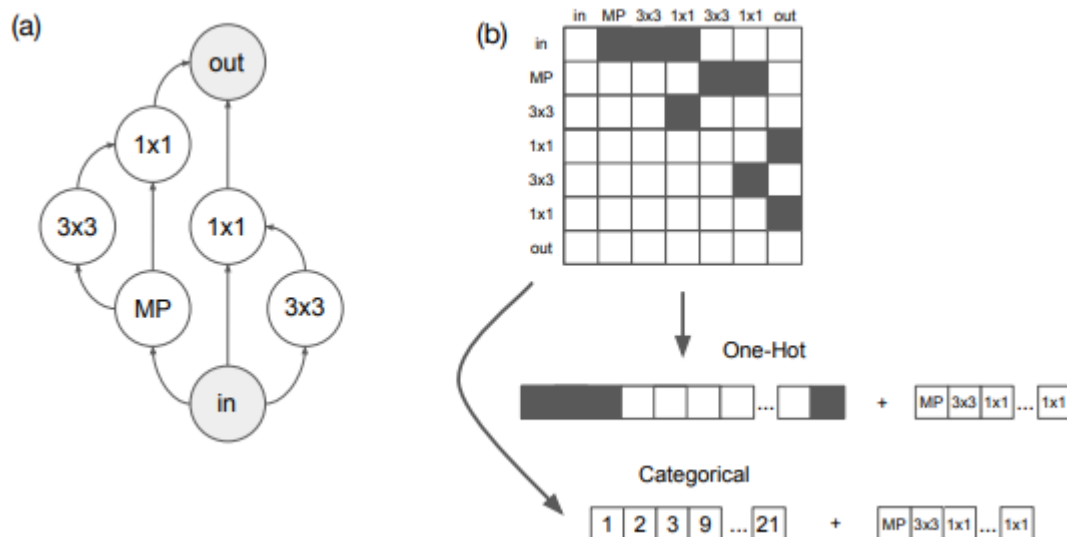


# Векторное представление сверточной нейронной сети

## Структура вектора



Картинка из статьи A Study on Encodings for Neural Architecture Search

## Связи

Сначала нам нужно описать связи между каждым узлом сети.

В качестве примера будем рассматривать блок модели resnet18.

Его структура для примера

- [0] Conv3x3
- [1] BatchNorm
- [2] ReLU
- [3] Conv3x3
- [4] BatchNorm
- [5] Skip-connection
- [6] Add() - [Слой 5] + [Слой 6]
- [7] ReLU

Нужно описать связи между слоями в блоке. Для этого построим матрицу смежности.

Матрица смежности  $A$  — это булева квадратная матрица размера  $n \times n$ , где  $n$  - количество слоев., а  $A[i][j] = 1$  - означает наличие связи между слоями.

Таким образом:

0 1 2 3 4 5 6 7 - Индексы слоев  
0 0 1 0 0 0 0 0 Conv3x3 - BatchNorm  
1 0 0 1 0 0 0 0 BatchNorm - ReLU

2 0 0 0 1 0 0 0 0 ReLU - Conv3x3  
3 0 0 0 0 1 0 0 0 Conv3x3 - BatchNorm  
4 0 0 0 0 0 1 0 0 BatchNorm - Add  
5 0 0 0 0 0 0 1 0 Skip-connection - Add  
6 0 0 0 0 0 0 0 1 Add - ReLU  
7 0 0 0 0 0 0 0 0 Конец блока

Матрица A символьная и разреженная, поэтому для векторного представления удобно использовать верхний треугольник без диагонали.

Векторизация (row-wise):

[1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 1 0 1] (верхняя часть без диагонали)

Это  $n(n-1)/2$  значений.

## Гарантия размера

Тут для гарантирования того, что итоговый вектор кодирования будет фиксированный длины мы задаем максимальное количество слоев. Таким образом, длина вектора всегда  $n\_max * (n\_max - 1)/2$ .

## Слои

Теперь кодируем какие конкретно операции применяются.

Тут можно использовать one-hot, тогда каждая операция будет закодирована как одномерная матрица или использовать просто кодирование числом.

Допустим:

Input = 0

Conv3x3 = 1

BatchNorm = 2

ReLU = 3

Output = 4

Так мы указываем тип слоя и дополнительно кодируем параметры каждого слоя.

Получаем например для Conv2d(kernel=3, stride=1, padding=1, out\_ch=64) вектор [1, 3, 1, 1, 64, 0, 0]

И получаем общий вектор всех слоев со всеми параметрами и их типами.

Каждый слой кодируется вектором фиксированной длины  $p$ .

Если всего  $n\_max$  слоев, получаем вектор размера  $n\_max \times p$ .

Если слой отсутствует — его вектор полностью из нулей (падинг).

Далее объединяем матрицу смежности и параметры слоев в один вектор.

Для этого просто конкатенируем вектора.

Вектор связей + вектор параметров слоев = итоговый вектор кодирования.

Итоговая длина:

$$L = n\_max(n\_max-1)/2 + n\_max \times p$$

**Для кодирования всей модели ResNet** нам нужно закодировать каждый блок, а после закодировать связи между блоками. Аналогично объединяем вектор блоков + вектор связей блоков = вектор всей нейросети  
Объединение векторов блоков и связей между блоками дает итоговый вектор архитектуры:

$L_{total} = \sum(L_{block\_i}) + L_{S\_block}$  для всех блоков  $i$  от 1 до  $MAX(k)$

Количество блоков должно быть конечным и ограниченным, для гарантии фиксированного размера.

**Для кодирования безблочной архитектуры** достаточно одной матрицы связей, которая объединиться с вектором слоев.

## Для восстановления

Разделяем вектор на:

- вектор связей
- вектор слоев

Т.к все имеет фиксированную длину, мы гарантированно знаем какой где.

Для каждого слоя:

Проверяем тип

Извлекаем параметры

Пропускаем слои, у которых все нули (это паддинг)

Восстанавливаем связи по матрице смежности.

## Проблемы

Номера узлов произвольны, то есть одна и та же архитектура может быть закодирована по-разному, если поменять местами узлы.

Это усложняет обучение, потому что одинаковые архитектуры могут выглядеть по-разному в векторной форме.

Обработка ошибок:

Если ошибка в связи - пропускаем связь как брак

Если ошибка в слое, задаем слою выбранные значения по умолчанию.