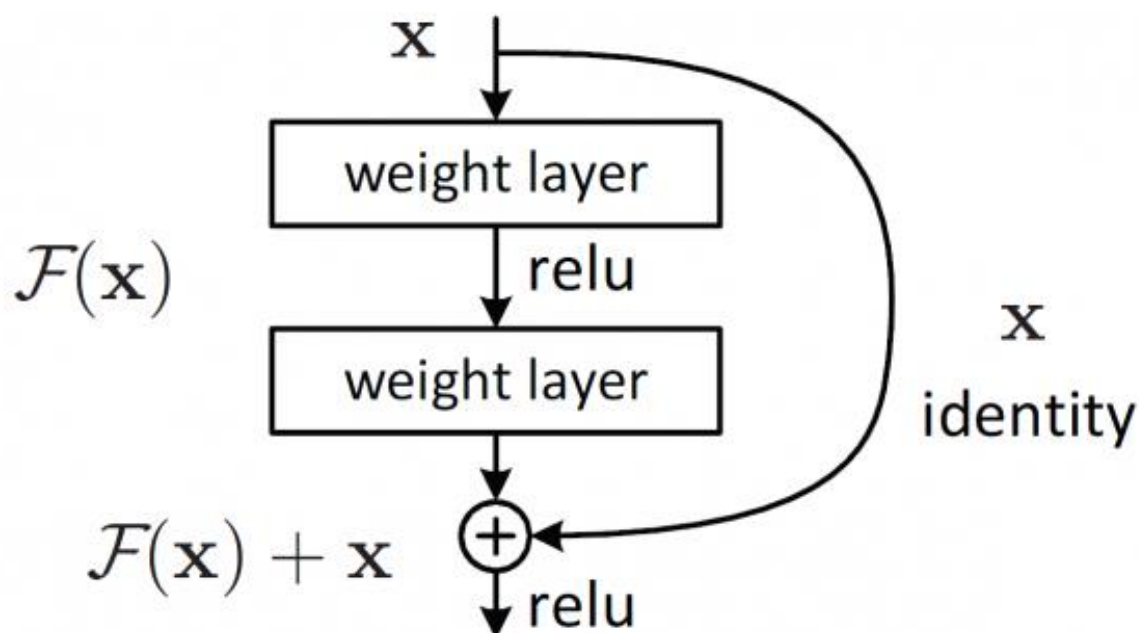


## 1. Что такое shortcut соединение и для чего оно нужно?

Когда более глубокая сеть начинает сворачиваться, возникает следующая проблема: с увеличением глубины сети точность сначала увеличивается, а затем быстро ухудшается («деградация»). Было сделано предположение, что если свёрточная нейронная сеть достигла своего предела точности на некотором слое, то все следующие слои должны будут вырождаться в тождественное преобразование, но из-за сложности обучения глубоких сетей этого не происходит. Чтобы преодолеть данную проблему Microsoft ввела глубокую «остаточную» структуру обучения. Вместо того, чтобы надеяться на то, что каждые несколько stacked layers непосредственно соответствуют желаемому основному представлению, они явно позволяют этим слоям соответствовать «остаточному». Формулировка  $F(x) + x$  может быть реализована с помощью нейронных сетей с соединениями для быстрого доступа.



Соединения быстрого доступа (shortcut connections) пропускают один или несколько слоев и выполняют сопоставление идентификаторов. Их выходы добавляются к выходам stacked layers. То есть нейронная сеть обучается предсказывать разницу между тем, что выдают предыдущие слои и таргетом, для компенсации этой разницы и вводится соединение быстрого доступа.

## 2. Что такое блок GRU?

Блок GRU – вентиляный рекуррентный блок, который используется в вентиляльных РНС. Один вентиляный блок одновременно управляет и коэффициентом забывания, и решением об обновлении блока и состояния.

Уравнения обновления имеют вид:

$$h_i^{(t)} = u_i^{(t-1)} h_i^{(t-1)} + (1 - u_i^{(t-1)}) \sigma \left( b_i + \sum_j U_{i,j} x_j^{(t-1)} + \sum_j W_{i,j} r_j^{(t-1)} h_j^{(t-1)} \right),$$

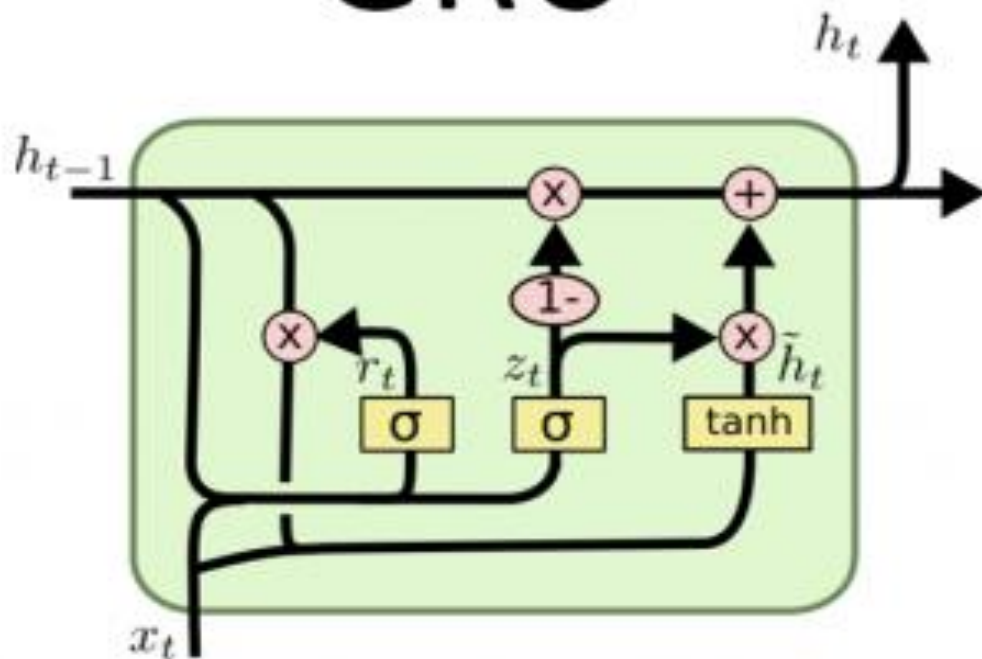
где  $x$  – входной вектор,  $h$  – вектор скрытого слоя, содержащий выходы всех ячеек, а  $b$ ,  $U$ ,  $W$  – соответственно смещения, веса входов и рекуррентные веса;  $u$  обозначает клапан «обновления»,  $r$  – клапан «сброса». Их значения определяются как обычно:

$$u_i^{(t)} = \sigma \left( b_i^u + \sum_j U_{i,j}^u x_j^{(t)} + \sum_j W_{i,j}^u h_j^{(t)} \right);$$
$$r_i^{(t)} = \sigma \left( b_i^r + \sum_j U_{i,j}^r x_j^{(t)} + \sum_j W_{i,j}^r h_j^{(t)} \right).$$

Клапаны обновления и сброса могут «игнорировать» части вектора состояния. Клапаны обновления действуют как условные интеграторы с утечкой с линейной функцией по любому измерению, т.е. могут либо скопировать вход, либо полностью проигнорировать его, заменив новым «целевым состоянием» (к которому интегратор с утечкой желает сойтись).

Клапаны сброса контролирует, какие части состояния использовать для вычисления следующего целевого состояния, и вносят дополнительный нелинейный эффект в соотношение между прошлым и будущим состояниями.

# GRU



### 3. Приведите ситуацию, когда в сети необходимо использовать несколько выходных слоев.

Например, в случае, когда нужно, чтобы нейронная сеть, получив на вход изображение с двумя геометрическими фигурами, на выходе отдала два изображения: на одном первый объект, а на другом второй.

Также в случае, если нужно получить на выходе изображения фона и объекта.

### 4. Как можно было бы повысить точность в данной задаче?

Для повышения точности в данной задаче можно было бы попробовать сделать ансамбль моделей (к примеру, Inception и ResNet).

Также можно увеличить количество эпох, что увеличит время необходимое для обучения сети и работы программы в целом.

Поиск наиболее удачных комбинаций слоёв и их параметров тоже может привести к увеличению точности.

**5. Объясните параметры в данной строке "Convolution2D (conv\_depth\_2, (kernel\_size, kernel\_size), padding='same', strides=(1, 1), activation='relu')(conv\_3)".**

- conv\_depth\_2 – целое число, размерность выходного пространства (то есть количество выходных фильтров в свертке).
- kernel\_size, kernel\_size – кортеж целых чисел, определяющих длину окна 1D свертки.
- padding='same' – приводит к заполнению ввода так, что длина вывода будет такой же, как у исходного ввода. (то есть возвращает то же измерение, что и входное изображение)
- strides=(1, 1) – кортеж целых чисел, определяющий длину шага свёртки.
- activation – функция активации, в нашей архитектуре мы использовали «relu».

**6. Для чего Вам нужны слои MaxPooling?**

Слой субдискретизации осуществляет уплотнение карт признаков предыдущего слоя и не изменяет количество карт. Каждая карта признаков слоя соединена с соответствующей картой признаков предыдущего слоя, каждый нейрон выполняет «сжатие» своего рецептивного поля посредством какой-либо функции. (Получает на вход маленькие отдельные фрагменты изображения (обычно 2x2) и объединяет каждый фрагмент в одно значение).

MaxPooling является одним из популярных видов слоя субдискретизации, из рецептивного слоя выбирается максимальное значение.

