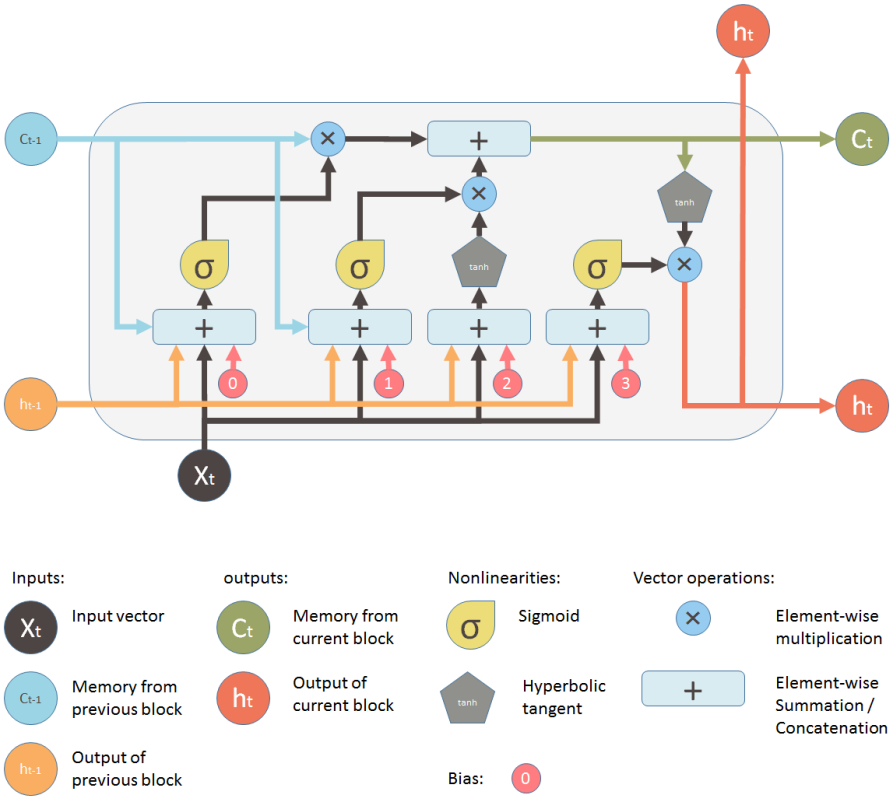
1.Что такое LSTM блок? Для чего нужен?

LSTM коротко не объяснить. Постараюсь выделить наиболее важные аспекты.

Некоторые задачи требуют определенной предыдущей информации, чтобы сделать какой-либо вывод. Пример: Пользователь пишет предложение, а мы должны угадать, какое слово подставить следующим. Контекст важен. И если мы будем обрабатывать предложение просто как “bad of words”, ничего разумного не выйдет. Соответственно приходим к тому, что нужно обрабатывать слова последовательно (RNN). Да, порядок важен, мы это учли, но нужна некоторая “память”, чтобы отражать контекст. Данную проблему решает LSTM блок. Приведу, конечно, картинку, но сильно описывать ее не буду.



На первый взгляд, жестоко, но на самом деле очень хитро и логично. На каждом шаге нам поступает на вход новая информация. Мы обрабатываем ее с помощью однослойной НС и “решаем” какую часть этой новой информации добавить к уже имеющейся. Уже имеющаяся информация тоже “фильтруется”, как клапаном. Мы можем забыть то, что у нас было,

а может прошлое нам все-таки важно. Отфильтрованные старая и новая информация сливаются в единое целое. Так же с помощью старого выхода, текущего входа и новой информации получается новый выход, который вместе с новой информацией передается на следующий шаг. (https://medium.com/mlreview/understanding-lstm-and-its-diagrams-37e2f46f1714)

2.Что такое сети ассоциативной памяти? Какое их применение?

Задачи:

а) Соотнести входную информацию со знакомыми объектами, и дополнить ее до точного описания объекта.

б) Отфильтровать из входной информации недостоверную, а на основании оставшейся решить первую задачу.

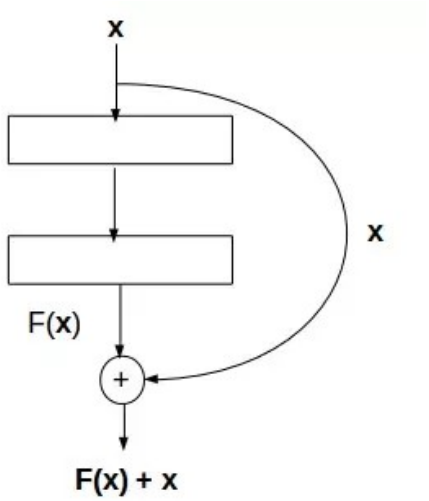
Как?

В случае автоассоциативной памяти входной и выходной вектор равны. Задача научить сеть запоминать паттерн. (энкодеры)

3.Что такое shortcut соединение и для чего оно нужно?

Коротко говоря, когда сеть становится очень глубокой появляется проблема “исчезающего” градиента и потеря точности (не из-за переобучения). Получается, что мелкая сеть справляется лучше глубокой, но недостаточно хорошо, а добавление слоев только ухудшает ситуацию. Решение residual NN. Будем передавать выход с текущего слоя не только следующему, но и еще слою, что лежит чуть глубже, допустим, через два от текущего. Проблема в том, что глубокая сеть имеет определенные проблемы с выучиванием тождественной функции, а мелкая справляется. Тогда в глубоких слоях будем пользоваться как минимум точностью полученной чуть раньше (shortcut) + чем-то, что пришло в текущий слой с предыдущего.

Мы учим сеть как будто другой функции (не той, что соответствует данным), но исходную легко восстановить. Схематично shortcut показан на картинке.

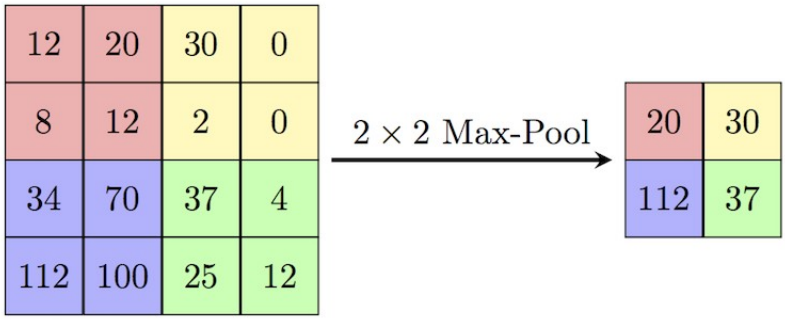


4.Всегда ли ядро свертки должно быть квадратным?

Нет, не всегда. Параметр kernel_size можно установить и прямоугольным. Грубо говоря, в таком случае мы будем “уделять больше внимания” определенной оси. Обычно, ядро все-таки квадратное и нечетного размера.

5.Что делает слой MaxPooling2D?

Алгоритмически:



Идеологически:

Max pooling выбирает признаки, которые наиболее существенны и сильнее влияют на выходное значение. Что такого “особенного” есть в текущем изображении.