

Распознавание объектов на фотографиях

1. Что такое LSTM блок? Для чего нужен?

Согласно принципиальной схеме LSTM, ячейки, рекуррентно связанные между собой, заменяют стандартные скрытые блоки в обыкновенных рекуррентных сетях. Входной признак вычисляется регулярным блоком искусственного нейрона. Его значение можно аккумулировать в состоянии, если сигмоидный входной клапан допускает это. В блоке состояния имеется линейная петля, вес которой управляется клапаном забывания. Выход ячейки можно перекрыть с помощью выходного клапана. Во всех клапанных блоках имеется сигмоидная нелинейность, тогда как во входном блоке разрешены произвольные сглаживающие нелинейности.

Вместо блока, который просто применяет поэлементную нелинейность к аффинному преобразованию входов и рекуррентным блокам, в рекуррентных LSTM-сетях имеются "LSTM-ячейки", обладающие внутренней рекуррентностью (петлей) в дополнение к внешней рекуррентности РНС.

У каждой ячейки такие же входы и выходы, как у обыкновенной рекуррентной сети, но ещё имеются дополнительные параметры и система клапанных блоков, управляющих потоком информации. Самым важным компонентом является блок состояния $s_i^{(t)}$ с линейной петлей, аналогичный описанному выше блоку с утечкой.

Вес петли управляется клапанным блоком забывания $f_i^{(t)}$, который присваивает этому весу значение от 0 до 1 с помощью сигмоиды:

$$f_i^{(t)} = \sigma(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)})$$

, где $x^{(t)}$ - текущий входной вектор, $h^{(t)}$ - вектор текущего скрытого слоя, содержащий выходы всех LSTM-ячеек, а b^f , U^f , W^f - соответственно смещения, веса входов и рекуррентные веса для клапанов забывания. Таким образом внутреннее состояние LSTM-ячейки обновляется по следующей формуле, в которой присутствует условный вес петли $f_i^{(t)}$:

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)})$$

, где b , U и W - соответственно смещения, веса входов и рекуррентные веса LSTM-ячейки.

Блок внешнего входного клапана $g_i^{(t)}$ вычисляется аналогично клапану забывания, но со своими параметрами:

$$g_i^{(t)} = \sigma(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)})$$

Выход $h_i^{(t)}$ LSTM-ячейки можно перекрыть с помощью выходного вентиля $q_i^{(t)}$, в котором также используется сигмоида:

$$h_i^{(t)} = \tanh(s_i^{(t)}) q_i^{(t)}$$

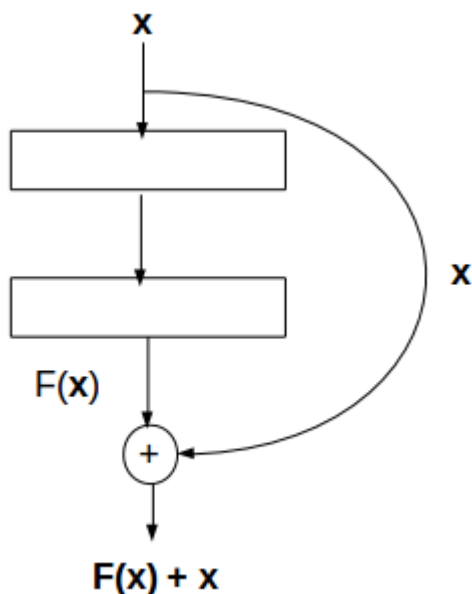
$$g_i^{(t)} = \sigma(b_i^0 + \sum_j U_{i,j}^0 x_j^{(t)} + \sum_j W_{i,j}^0 h_j^{(t-1)})$$

LSTM-блоки - решение проблемы краткосрочной памяти в РНС, способные обучаться долгосрочным зависимостям.

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

В лекционном материале не нашёл ответа на данный вопрос. В соответствии со [страницей](#):

Вместо того, чтобы ожидать, что каждые несколько сложенных слоев будут непосредственно соответствовать желаемому основному отображению, мы разрешаем этим слоям соответствовать остаточному отображению. Пусть $H(x)$ - желаемое основное отображение. Мы пытаемся сделать сложенные нелинейные слои подходящими для другого отображения $F(x) = H(x) - x$. Тогда исходное отображение можно преобразовать в $F(x) + x$.



Сокращенные соединения - это соединения, которые пропускают один или несколько слоев. На рисунке добавлен ярлык идентификации к изученной остаточной карте $F(x)$ для получения желаемого отображения $H(x)$. Чрезвычайно глубокие остаточные сети легче оптимизировать, чем их обычные сети. Кроме того, остаточные сети решают проблему ухудшения качества и дают лучшую производительность, чем обычные сети.

Сокращенные соединения имеют дополнительное преимущество: они не добавляют никаких дополнительных параметров или сложности вычислений.

3. Чем можно обосновать, что при меньшем ядре свертки результат лучше?

Чем меньше ядро свёртки, тем меньше информации об исходном изображении мы теряем во время свёртки. Обратной стороной медали является тот факт, что увеличивается объём памяти и время необходимые для обучения сети. Поскольку размер исходных изображений небольшой, то мы можем позволить себе использовать использовать ядро небольшого размера. Да и вообще, стоит выбирать размер ядра как можно меньшим в большинстве случаев.

4. Для чего в Вашей модели нужен слой Flatten?

Вроде простой вопрос. Объясняется особенностями свёрточных нейронных сетей и перцептрона.

Выход свёрточной нейронной сети (карту признаков) развернуть в вектор для того, что можно было скормить её перцептрону. Перцептрон векторы принимает на вход, поэтому без Flatten он умрёт.