

Нейросети. Задание 1. Теория

Шеншиной Юлии, 491

April 18, 2017

Вопрос 1: Чем нейросети отличаются от линейных моделей а чем похожи?

Ответ: Нейросети состоят из линейных и нелинейных слоев. Линейные слои осуществляют линейные преобразования пространства, разделяя его. Нелинейные слои являются отличием нейросетей от линейных моделей и позволяют построить более сложную модель.

Вопрос 2: В чем недостатки полносвязных нейронных сетей какая мотивация к использованию сверточных?

Ответ: Полносвязные нейронные сети "запоминают" гораздо больше информации - каждый пиксель имеет свои настраиваемые веса, как следствие обучение проходит сильно дольше. Сверточные нейронные сети имеют намного меньше настраиваемых весов и, объединяя блоки пикселей в единый объект, запоминают небольшие паттерны. Вследствие чего обучение происходит быстрее, более того, сети становятся менее чувствительны к расположению объекта.

Вопрос 3: Какие слои используются в современных нейронных сетях? Опишите как работает каждый слой и свою интуицию зачем он нужен.

Ответ:

- DenseLayer – линейное преобразование $Wx + b$, можно интерпретировать как разделение пространства объектов гиперплоскостями.
- Сверточный слой – основной блок сверточной сети. Фильтр представляет собой матрицу, которая поэлементно перемножается с подматрицами исходной матрицы. Результаты суммируются (внутри блока) и записываются в новую матрицу – выходные данные слоя. В результате сеть ищет паттерны, состоящие из более чем одного пикселя.
- Rectified linear unit (ReLU) – слой активации, позволяющий добавить нелинейность.
- Пуллинг – нелинейное уплотнение слоев (например, из квадрата 2×2 выбирается среднее или максимум). Позволяет сильно уменьшить размер входных данных + в случае, когда мы точно знаем, что слишком подробное исследование признаков на данном этапе не нужно, позволяет не переобучиться.
- DropLayer – случайным образом выбрасываем некоторые нейроны. Когда хорошо? (Пример с лекции) Хоти найти кота на изображении. Кот – объект с ушами, носом и глазами. Рассматриваем такую

картинку:



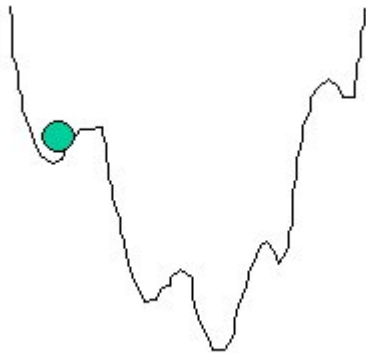
Не видно уши. Значит объект – не кот. Если дропнуть некоторые случайные нейроны, то с большой вероятностью мы опознаем кота.

Вопрос 4: Может ли нейросеть решать задачу регрессии, какой компонент для этого нужно заменить в нейросети из лекции 1?

Ответ: Да, может.

Вопрос 5: Почему обычные методы оптимизации плохо работают с нейросетями? А какие работают хорошо? Почему они работают хорошо?

Ответ: Обычные методы оптимизации типа GB работают плохо из-за множества локальных минимумов минимизируемой функции ошибки (см. рисунок):



попав в локальный минимум, мы уже не покинем его.

Вместо этого используем: Adam, RMSProp, Nesterov Momentum. Все они используют метод скользящего среднего, то есть каждый новый шаг делается не в направлении градиента, а в сторону $\alpha \cdot grad_i + \beta grad_{i+1}$.

Вопрос 6: Для чего нужен backprop, чем это лучше/хуже чем считать градиенты без него? Почему backprop эффективно считается на GPU?

Ответ: Backprop – алгоритм обратного распространения ошибки. Позволяет быстро считать градиенты сложных функций с использованием входных значений. Хорошо: не нужно на каждом узле считать сложные производные по большому числу параметров, они вычисляются на основе подсчитанной один раз производной на последнем из уровней \Rightarrow быстро и можно распараллелить вычисления. Плохо: минимизируем ошибку методом градиентного спуска \Rightarrow можем попасть в локальный минимум и никогда уже из него не выбраться, хотя рядом будет намного более оптимальное решение.

Вопрос 7: Почему для нейросетей не используют кросс валидацию, что вместо неё? Можно-ли ее использовать? **Ответ:** Нейросети слишком долго учатся, поэтому кросс-валидацию не используют.

Вопрос 8: Ответ:

padding	stride	width	width	depth
'same'	1	28	28	8
'valid'	1	26	26	8
'valid'	2	13	13	8