Школа анализа данных
Машинное обучение, часть 2
Теоретическое задание №1

Кошман Дмитрий

Задача 1

1. Batch Normalization

$$f(x) = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}}$$

Где μ, σ^2 - выборочное среднее и вариация по признакам для данного батча.
Запишем подробнее, пусть x_{ij} – j признак i элемента.

$$f_{ij}(x) = \frac{x_{ij} - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}},$$

$$\mu_j = \frac{1}{n}\sum_k x_{kj}$$

$$\sigma_j^2 = \frac{1}{n}\sum_k (x_{kj} - \mu_j)^2$$

Обозначим функцию потерь L. Тогда задача звучит так: нам дан градиент ∇L_f, нужно найти ∇L_x.

$$\frac{\partial \mu_j}{\partial x_{ij}} = \frac{1}{n}$$

$$\frac{\partial \sigma_j^2}{\partial x_{ij}} = \frac{2}{n}\sum_k (x_{kj} - \mu_j)([k = i] - \frac{1}{n}) = \frac{2}{n}(x_{ij} - \mu_j) - \frac{2}{n^2}\sum_k (x_{kj} - \mu_j) = \frac{2}{n}(x_{ij} - \mu_j)$$

$$(\nabla L_x)_{ij} = \frac{\partial L}{\partial x_{ij}} = \sum_{kl} \frac{\partial L}{\partial f_{kl}} \frac{\partial f_{kl}}{\partial x_{ij}} = \sum_k \frac{\partial L}{\partial f_{kj}} \frac{\partial f_{kj}}{\partial x_{ij}} =$$

$$= \sum_k (\nabla L_f)_{kj} \left(\frac{[k = i] - \frac{1}{n}}{\sqrt{\sigma_j^2 + \epsilon}} - \frac{(x_{kj} - \mu_j)(x_{ij} - \mu_j)}{n(\sigma_j^2 + \epsilon)^{3/2}} \right)$$

2. Dropout

$$f_{ij}(x) = \frac{1}{1-p}(1 - b_{ij})x_{ij}, \;\; b_{ij} \sim Bern(p)$$

$$(\nabla L_x)_{ij} = (\nabla L_f)_{ij} \frac{1 - b_{ij}}{1 - p}$$

3. Conv2d

$$f_{bc_{out}hw}(x) = \sum_{x,y\in[-kernelsize,kernelsize],c_{in}} \hat{x}_{bc_{in}(h+x)(w+y)} k_{c_{out}c_{in}xy} + \text{bias}_{c_{out}}$$

$$\hat{x} - \text{padded } x, k - \text{kernels}$$

$$(\nabla L_x)_{bc_{in}hw} = \sum_{B,c_{out},H,W} \frac{\partial L}{\partial f_{Bc_{out}HW}} \frac{\partial f_{Bc_{out}HW}}{\partial x_{bc_{in}hw}} =$$

$$\sum_{c_{out},H,W} \frac{\partial L}{\partial f_{bc_{out}HW}} \frac{\partial f_{bc_{out}HW}}{\partial x_{bc_{in}hw}} =$$

$$\sum_{x,y\in[-kernelsize,kernelsize],c_{out}} \frac{\partial L}{\partial f_{bc_{out}(h-x)(w-y)}} k_{c_{out}c_{in}xy}$$

Можно заметить, что это свертка выходного градиента относительно тех же весов с переставленными каналами и повернутыми на 180 ядрами.

$$(\nabla L_k)_{c_{out}c_{in}xy} = \sum_{b,C_{out},H,W} \frac{\partial L}{\partial f_{bC_{out}HW}} \frac{\partial f_{bC_{out}HW}}{\partial k_{c_{out}c_{in}xy}} =$$

$$\sum_{b,H,W} \frac{\partial L}{\partial f_{bc_{out}HW}} \frac{\partial f_{bc_{out}HW}}{\partial k_{c_{out}c_{in}xy}} =$$

$$\sum_{h,w\in\{(h+x,w+y)\in f\},b} \frac{\partial L}{\partial f_{bc_{out}(h+x)(w+y)}} x_{bc_{in}(h+x)(w+y)}$$

Это тоже можно представить в виде свертки выходного градиента относительно входа, если потом еще повернуть полученные изображения на 180.

$$(\nabla L_{\text{bias}})_{c_{out}} = \sum_{b,h,w} (\nabla L_f)_{bc_{out}hw}$$

Задача 2

MobileNet

1. Какую задачу решают авторы?

Авторы предлагают новую легковесную и эффективную архитектуру нейросети для компьютерного зрения под названием MobileNet. Они решили заполнить нишу спроса, где скорость стоит на первом месте и существуют сильные ограничения на доступную вычислительную мощность. В первую очередь они видят применение MobileNet на мобильных устройствах для решения задач узнавания, детекции и классификации объектов на изображениях. В общем авторы хотят предоставить максимально практичную нейросеть для данных задач.

2. Какова основная идея предлагаемого авторами решения поставленной задачи?

Для достижения эффективности авторы решили факторизовать проверенный опытом сверточный слой на два отдельных - depthwise и pointwise слои.

Структура по слоям :

$$DepthwiseConv \to BatchNorm \to ReLU \to PointwiseConv \to BatchNorm \to ReLU$$

Depthwise слой имеет ядро размера $in\_channels \times K \times K$ и разбивает входное изображение размера $batch \times in\_channels \times H \times W$ на группы по каналам и применяет к каждой группе обычную одномерную свертку, затем конкатенирует результаты.

Pointwise слой – это обычная свертка с ядром $out\_channels \times in\_channels \times 1 \times 1$

В таком виде количество вычислений падает с $H \cdot W \cdot in\_channels \cdot out\_channels \cdot K^2$ до $H \cdot W \cdot in\_channels \cdot (K^2 + out\_channels)$, то есть примерно в K^2 раз.

А для достижения практичности авторы предоставляют два гиперпараметра, дающих контроль над скоростью и размером сети за счет регулирования количества каналов в скрытых слоях и сжатия исходного изображения.

Почему это работает, несмотря на огрубление тензорных операций? Потому что здесь используется тот же принцип, за счет которого свертка и показывает хорошие результаты - поиск pattern в изображении и агрегирование результатов поиска по всем каналам в новые, более абстрактные признаки.

3. Каковы результаты, полученные авторами?

1. На данных ImageNet: по сравнению с обычной сверткой получили ускорение и уменьшение в пять раз за счет падения точности на 1%
2. Экспериментально показали, что глубокая сеть работает немного лучше, чем широкая с тем же количеством параметров.
3. Показали, что качество модели лог-линейно зависит от количества базовых операций.
4. Добились одинаково качества с моделью $VGG16$, затрачивая в 30 раз меньше времени и памяти.
5. Показали незначительное отставание от лучших моделей за счет значительно уменьшения затрат по задачам геолокации, узнавания лиц, детекции.

Задача 3 SphericalCNN

1. Какую задачу решают авторы?

Авторы хотят, чтобы в арсенале машинного обучения были эффективные методы работы со сферическими изображениями, которые возникают в таких задачах, как зрение роботов и глобальное предсказание погоды.

Сверточные сети, которые хорошо справляются с аналогичными задачами для изображений в прямолинейных координатах, плохо переносятся на сферические изображения, так как не существует изоморфного преобразования для дискретных представлений соответствующих изображений. Поэтому возникает желание перенести принцип, благодаря которому работают сверточные сети - инвариантность класса относительно движений плоскости - на изображения в сферических координатах.

Такое обобщение можно сделать с помощью понятия эквивариантности - нахождения таких эквивариантных преобразований, которые перестановочны с интересуемой нас группой симметрии. В случае с двумерными изображениями таковыми являются свертки относительно движений. Хочется найти соответсвующие сферические свертки относительно ортогональной группы преобразований, сохраняющих ориентацию.

2. Какова основная идея предлагаемого авторами решения поставленной задачи?

Сначала вводится понятие сферической корреляции и корреляции группы ротаций, а потом полученные чисто математические выводы переносятся на практическую реализацию в виде нейронной сети со сферическими свертками.

Как понимать сферическую корреляцию и зачем она нужна? Для двух фиксированных сферических изображений их сферическая корреляция - это функция, отображающая элемент группы ротации в меру похожести изображений, если к первому применить данную ротацию. Она интересна нам тем, что если два изображения содержат один и тот же объект в различных ориентациях, то максимум сферической корреляции достигается в ротации, которая соответствует разнице между истинными ориентациями. Корреляция группы ротаций это то же самое, но не для изображений, а для ротаций. То есть ее максимум для ротаций R_1 и R_2 достигается в ротации, переводящей R_1 в R_2. К обеим корреляциями можно применить градиентные методы, позволяющие приближенно находить эти максимумы. Причем если изображения содержат одинаковый объект, то сферическая корреляция будет иметь большой пик в точке, соответствующей истинной ротации, что позволит точно определить, что изображения из одного класса.

Как преобразовывать выход сферической свертки в виде числа? Выход слоя сферической свертки - элемент группы $SO(3)$, который дискретно представим в трехмерных угловых координатах Эйлера, то есть обычного тензора, и здесь работают те же методы агрегации, как для обычной свертки.

Наивная имплементация ротационной свертки будет работать с трехмерными матрицами, то есть иметь сложность порядка $O(n^6)$. Авторы показали, что элементы группы ротаций удовлетворяют требованиям теоремы Фурье, что позволяет представить корреляцию в виде скалярного произведения трансформированных преобразований. Используя это, и возможность вычисления преобразования Фурье за $O(n \log n)$, общая сложность падает до $O(n^4 \log^2 n)$.

3. Каковы результаты, полученные авторами?

1. Авторы показали, что практически воплощенные модели отличаются от чистых математических на погрешность, растущую логарифмически относительно разрешения изображения и линейно относительно глубины модели.

2. На данных MNIST, не подвергнутых ротации, сферические свертки показывали такое же качество, как обычные. На повернутых же изображениях сферические свертки сохранили качество, тогда как обычные работали не лучше случайного предсказания.

3. На данных ShapeNet сферические свертки показали качество немного хуже лидеров, причем не потребовав точного подбора параметров и ad-hoc изменений в архитектуре.

4. В задаче QM7 сферические свертки хорошо предсказали потенциальную энергию молекулы, уступив по качеству только лидеру, которому потребовалась существенная предобработка данных.