Тятя! Тятя! Нейросети заменили продавца!

Ппилиф Ульянкин

https://github.com/FUlyankin/neural_nets_prob

Листочек 5: алгоритм обратного распространения ошибки

> К толковому выбору приводит опыт,а к нему приводит выбор бестолковый.
>
> *JSON Стэтхэм*

Упражнение 1 (граф вычислений)

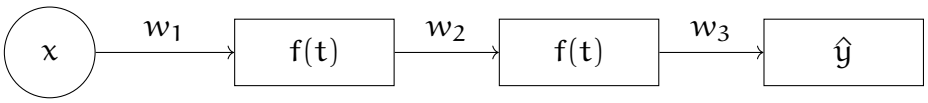
Как найти производную a по b в графе вычислений? Находим не посещённый путь из a в b, перемножаем все производные на рёбрах получившегося пути. Добавляем это произведение в сумму. Так делаем для всех путей. Маша хочет попробовать этот алгоритм на функции

$$f(x, y) = x^2 + xy + (x + y)^2.$$

Помогите ей нарисовать граф вычислений и найти $\frac{\partial f}{\partial x}$ и $\frac{\partial f}{\partial y}$. В каждой вершине графа записывайте результат вычисления одной элементарной операции: сложений или умножения[1].

Упражнение 2 (придумываем backpropagation)

У Маши есть нейросеть с картинки ниже, где w_k — веса для k слоя, $f(t)$ — какая-то функция активации. Маша хочет научиться делать для такой нейронной сетки градиентный спуск.



а. Запишите Машину нейросеть, как сложную функцию.

б. Предположим, что Маша решает задачу регрессии. Она прогоняет через нейросетку одно наблюдение. Она вычисляет знчение функции потерь $L(w_1, w_2, w_3) = \frac{1}{2} \cdot (y - \hat{y})^2$. Найдите производные функции L по всем весам w_k.

[1] По мотивам книги Николенко "Глубокое обучение"(стр. 79)

в. В производных постоянно повторяются одни и те же части. Постоянно искать их не очень оптимально. Выделите эти части в прямоугольнички цветными ручками.

г. Выпишите все производные в том виде, в котором их было бы удобно использовать для алгоритма обратного распространения ошибки, а затем, сформулируйте сам алгоритм. Нарисуйте под него удобную схемку.

Упражнение 3 (сигмоида)

В **не**глубоких сетях в качестве функции активации можно использовать сигмоиду

$$\sigma(z) = \frac{1}{1+e^{-z}} = \frac{e^z}{1+e^z},$$

Маша хочет использовать сигмоиду внутри нейросети. Предполагается, что после прямого шага, наши вычисления будут использованы в другой части нейросети. В конечном итоге, по выходу из нейросети мы вычислим какую-то функцию потерь L.

У сигмоиды нет параметров. Чтобы обучить нейросеть, Маше понадобится производная $\frac{\partial L}{\partial z}$. Выпишите её в матричном виде через производные $\frac{\partial L}{\partial \sigma}$ и $\frac{\partial \sigma}{\partial z}$.

Упражнение 4 (линейный слой)

Маша знает, что главный слой в нейронных сетях — линейный. В матричном виде его можно записать как $Z = XW$.

Маша хочет использовать этот слой внутри нейросети. Предполагается, что после прямого шага наши вычисления будут использованы в другой части нейросети. В конечном итоге, по выходу из нейросети мы вычислим какую-то функцию потерь L.

Чтобы обучить нейросеть, Маше понадобятся производные $\frac{\partial L}{\partial X}$ и $\frac{\partial L}{\partial W}$. Аккуратно найдите их и запишите в матричном виде[2]. Предполагается, что

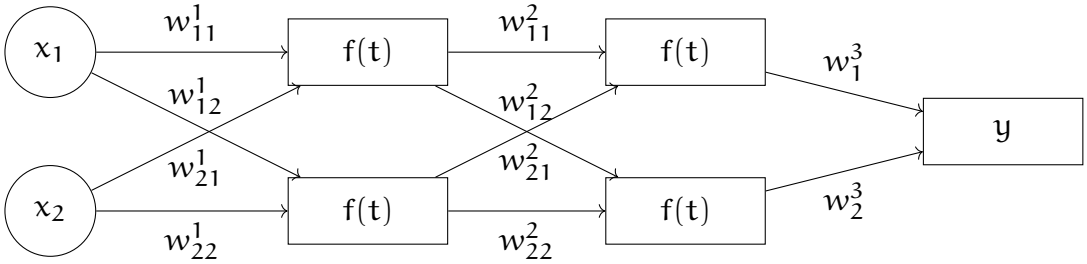
$$X = \begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{pmatrix} \qquad W = \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{pmatrix}$$

$$Z = XW = \begin{pmatrix} z_{11} & z_{12} & z_{13} \\ z_{21} & z_{22} & z_{23} \end{pmatrix} = \begin{pmatrix} x_{11}w_{11} + x_{12}w_{21} & x_{11}w_{12} + x_{12}w_{22} & x_{11}w_{13} + x_{12}w_{23} \\ x_{21}w_{11} + x_{22}w_{21} & x_{21}w_{12} + x_{22}w_{22} & x_{21}w_{13} + x_{22}w_{23} \end{pmatrix}$$

Упражнение 5 (Backpropagation в матричном виде)

У Маши есть нейросеть с картинки ниже, где w_{ij}^k — веса для k слоя, $f(t)$ — какая-то функция активации. Маша хочет научиться делать для такой нейронной сетки градиентный спуск.

[2] https://web.eecs.umich.edu/~justincj/teaching/eecs442/notes/linear-backprop.html



а. Запишите Машину нейросеть, как сложную функцию. Сначала в виде нескольких уравнений, а затем в матричном виде.

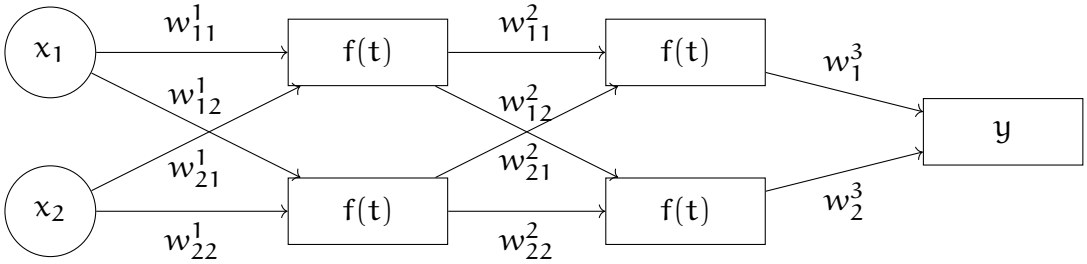
б. Выпишите все производные в том виде, в котором их было бы удобно использовать для алгоритма обратного распространения ошибки, а затем, сформулируйте сам алгоритм. Нарисуйте под него удобную схемку.

Упражнение 6 (Backpropagation своими руками)

У Маши есть нейросеть с картинки ниже. Она использует функцию потерь

$$L(W_1, W_2, W_3) = \frac{1}{2} \cdot (\hat{y} - y)^2.$$

В качестве функции активации Маша выбрала сигмоиду $\sigma(t) = \frac{e^t}{1+e^t}$.



Выпишите для Машиной нейросетки алгоритм обратного распространения ошибки в общем виде. Пусть Маша инициализировала веса нейронной сети нулями. У неё есть два наблюдения

№	x_1	x_2	y
1	1	1	1
2	5	2	0

Сделайте руками два шага алгоритма обратного распространения ошибки. Пусть скорость обучения $\eta = 1$. Стохастический градиентный спуск решил, что сначала для шага будет использоваться второе наблюдение, а затем первое. Объясните, почему инициализировать веса

нулями — плохая идея. Почему делать инициализацию весов любой другой константой — плохая идея?

Упражнение 7 (Незаметный backpropagation)

Маша собрала нейросеть:

$$y = \max\left(0; X \cdot \begin{pmatrix} 1 & -1 \\ 0.5 & 0 \end{pmatrix}\right) \cdot \begin{pmatrix} 0.5 \\ 1 \end{pmatrix}$$

Теперь Маша внимательно смотрит на неё.

а) Первый слой нашей нейросетки — линейный. По какой формуле делается forward pass? Сделайте его для матрицы

$$X = \begin{pmatrix} 1 & 2 \\ -1 & 2 \end{pmatrix}.$$

б) Найдите для первого слоя производную выхода по входу. При обратном движении по нейросетке, в первый слой пришёл накопленный градиент

$$d = \begin{pmatrix} -0.5 & 0 \\ 0 & 0 \end{pmatrix}.$$

Каким будет новое накопленное значение градиента, которое выплюнет из себя линейный слой? По какой формуле делается backward pass?

в) Второй слой нейросетки — функция активации, ReLU. По какой формуле делается forward pass? Сделайте его для матрицы

$$H_1 = \begin{pmatrix} 2 & -0.5 \\ 0 & 1 \end{pmatrix}.$$

г) Найдите для второго слоя производную выхода по входу. При обратном движении по нейросетке во второй слой пришёл накопленный градиент

$$d = \begin{pmatrix} -0.5 & -1 \\ 0 & 0 \end{pmatrix}.$$

Каким будет новое накопленное значение градиента, которое выплюнет из себя ReLU? По какой формуле делается backward pass?

д) Третий слой нейросетки — линейный. По какой формуле делается forward pass? Сделайте его для матрицы

$$O_1 = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}.$$

е) Найдите для третьего слоя производную выхода по входу. При обратном движении по нейросетке, в третий слой пришёл накопленный градиент $d = (-1, 0)^T$. Каким будет новое накопленное значение градиента, которое выплюнет из себя линейный слой?

ж) Мы решаем задачу Регрессии. В качестве функции ошибки мы используем

$$MSE = \frac{1}{2n} \sum (\hat{y}_i - y_i)^2.$$

Пусть для рассматриваемых наблюдений реальные значения $y_1 = 2, y_2 = 1$. Найдите значение MSE.

з) Чему равна производная MSE по прогнозу? Каким будет накопленное значение градиента, которое MSE выплюнет из себя в предыдущий слой нейросетки?

и) Пусть скорость обучения $\gamma = 1$. Сделайте для весов нейросети шаг градиентного спуска.

к) Посидела Маша, посидела, и поняла, что неправильно она всё делает. В реальности перед ней не задача регрессии, а задача классификации. Маша применила к выходу из нейросетки сигмоиду. Как будет для неё выглядеть forward pass?

л) В качестве функции потерь Маша использует logloss . Как для этой функции потерь выглядит forward pass? Сделайте его.

м) Найдите для logloss производную прогнозов по входу в сигмоиду. Как будет выглядеть backward pass, если $y_1 = 0, y_2 = 1$? Как поменяется оставшаяся часть алгоритма обратного распространения ошибки?

Упражнение 8 (Нестеров и backprop)

К Маше приехал её папа и загрузил её интересным вопросом. В алгоритме обратного распространения ошибки мы можем делать шаг как минимум двумя способами:

а. Зафиксировали все w_{t-1}, нашли все градиенты, сделали сразу по всем весам шаг градиентного спуска.

б. Нашли градиенты для последнего слоя и сделали шаг для его весов, получили w_t^k. Для поиска градиентов предпоследнего слоя используем веса w_t^k, а не w_{t-1}^k. Все остальные слои обновляем по аналогии.

Как думаете, какой из способов будет приводить к более быстрой сходимости и почему[3]?

[3] Я придумал эту задачу и не смог найти статью, где делали бы что-то похожее. Если вы видели такую, пришлите мне её плиз.