Применение методов квантования к решению задачи оптимизации нейронных сетей

Шлыков Илья

2022

Введение

Квантование нейронной сети - метод преобразования данных, который позволяет уменьшить размер модели и получить преимущество в скорости работы, при небольшой потери качества.

Симметричное равномерное квантование

Самым простым методом среди существующих можно считать метод симмметричного равномерного квантования. Для некоторых данных X, которые изменяются в диапозоне (l,u) и некоторого значения среза $C \in (0, max(|l|, |u|))$ симметричное квантование к int8 может быть сформированно как:

$$q = round(clip(x, c)/s) \tag{1}$$

где $clip(x, c) = min(max(x, -c), c), s = 2 * c/(2^8 - 1)$ в свою очередь s это коэффициент масштабирования для проецирования чисел с запятой в 8-битное целое число. Деквантованые веса могут быть вычеслены как: $x = q * s$

Аффинный квантователь

Более общим методом является метод аффинного квантователя, который отличается от предыдущего метода, тем что помимо параметра s, появляется новый параметр Z(zero-point), который имеет тот же тип, что и квантованные значения q, и является квантованным значением, соответствующим действительному значению 0. Таким образом гарантируется, что 0 будет представлен среди квантованных значений.После того как параметры определены, процесс квантования может быть представлен как:

$$x_{int} = round(\frac{x}{s}) + Z \tag{2}$$

$$q = clip(0, N_{levels} - 1, x_{int}) \tag{3}$$

Деквантование: $x = (q - Z) * s$

Стохастический квантователь

Стохастическое квантование моделирует аддитивный шум, с последующим округлением. Задается следующим образом:

$$x_{int} = round(\frac{x + \epsilon}{S}) + Z, \epsilon \ Unif(-1/2, 1/2) \tag{4}$$

$$q = clip(0, N_{levels} - 1, x_{int}) \tag{5}$$

Деквантование происходит аналогично деквантованию в аффинном квантователе

Квантование во время обучения

При квантовании после обучения, есть вероятность сильно потерять в точности так как обучалась сеть с неквантованными значениями. Для того, чтобы повысить точность можно применить метод квантования во время обучения, за счет того, что сеть будет обучаться используя уже проквантованные веса.

Learned step size quantization (LSQ)

LSQ метод основан на s масштабировании, но также использует градиент,который вычисляется используя straight through estimator(STE), для корректировки размера шага.

$$y = w_q * x + b$$

$$w_q = [w/s] * s$$

$$\delta y/\delta s = \frac{\delta y}{\delta w_q} * \frac{\delta w_q}{\delta s}$$

$$\frac{\delta w_q}{\delta s} = s\delta/\delta s([w/s]) + [w/s]$$

Так как функция [w/s] не дифференцируемая используем STE

$$\delta/\delta s([w/s]) = \delta/\delta s(w/s) = -w/s^2$$

Learned step size quantization (LSQ)

Итоговый результат:

$$\frac{\delta w_q}{\delta s} = \begin{cases} -w/s + [w/s] \quad if \quad -Q_N < w/s < Q_P \\ -Q_N \quad if \quad w/s < -Q_N \\ Q_P \quad if \quad w/s > Q_P \end{cases} \tag{6}$$

Корректировка весов будет выглядеть следующим образом:

$$w_{float} = w_{float} - \mu \delta L / \delta w_{out} * I_{w_{out}} \in (w_{min}, w_{max})$$