О квантовании нейросетей

Салищев Сергей

Нейросеть

- Непрерывный и гладкий объект

$$x_{i+1} = f(w_i x_i + b_i)$$
$$x_i \in R^{n_i}$$
$$f \in \mathrm{C}^1$$

- Обучается градиентным спуском

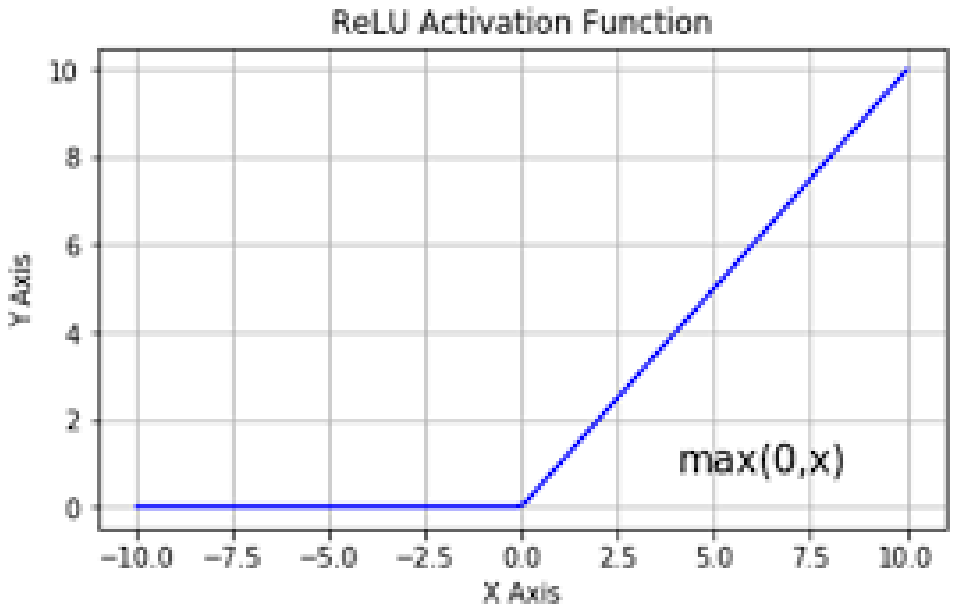
Пусть $L(\theta) \to min$ - целевая функция, γ – скорость обучения

$$g_t = \nabla_\theta L(\theta_{t-1})$$
$$\theta_t = \theta_{t-1} - \gamma_t g_t$$

Теряем гладкость

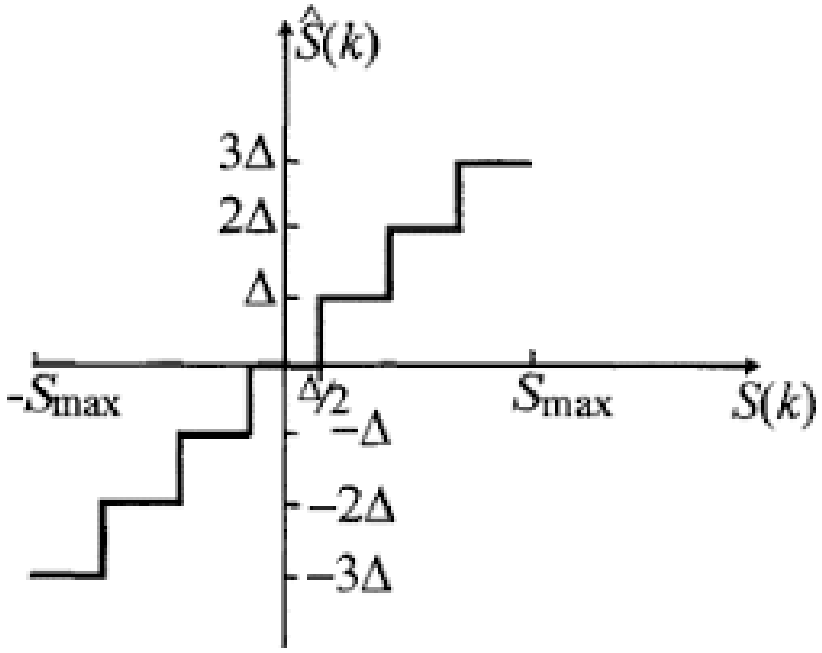
- ReLU
- Сеть с ReLU эквивалентна дереву решений*

*Aytekin C. Neural Networks are Decision Trees //arXiv preprint arXiv:2210.05189. – 2022.

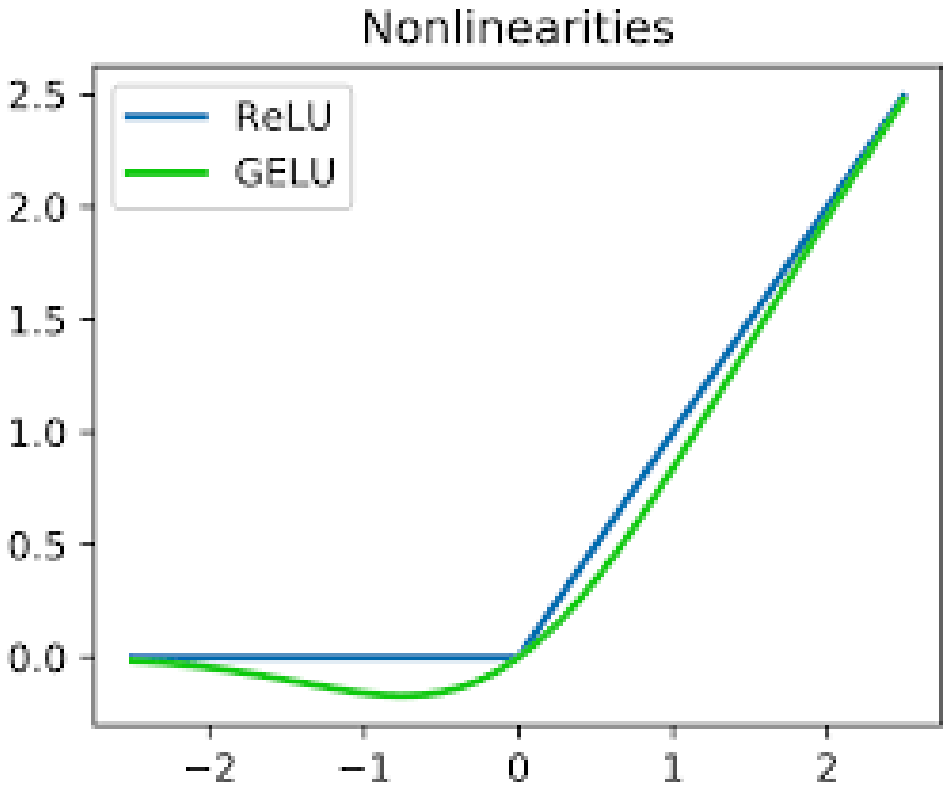


Теряем непрерывность

- Вычисления с фиксированной точкой малой разрядности
  - Сокращение длины арифметического переноса между разрядами
- Спайковые сети
  - Моделирование биологических нейросетей
- Вычисления в памяти
  - Нейросети на чипе
  - Физическая нечеткая логика



В поисках потерянной гладкости



Градиентный спуск? Контрастное обучение

- В естественных нейросетях градиентный спуск (пока) не найден*
- Как тренировать нейросети без ГС*?
  - Послойное контрастное обучение
  - Как научиться попадать в мишень
    1. Научится попадать кучно
    2. Сдвинуть прицел на центр мишени
  - Мешок признаков
    - Сначала генерируем признаки, потом вешаем метки
    - Вероятность линейного разделения кластеров → 1 с ростом числа признаков
- Градиентный спуск (в разы) быстрее
- Обучение нейросетей в природе интерпретируется как контрастное обучение

*Hinton G. The forward-forward algorithm: Some preliminary investigations //arXiv preprint arXiv:2212.13345. – 2022.

Пакетный Стохастический градиент

$$L(w) = \frac{1}{n}\sum_i L_i\,(w)$$

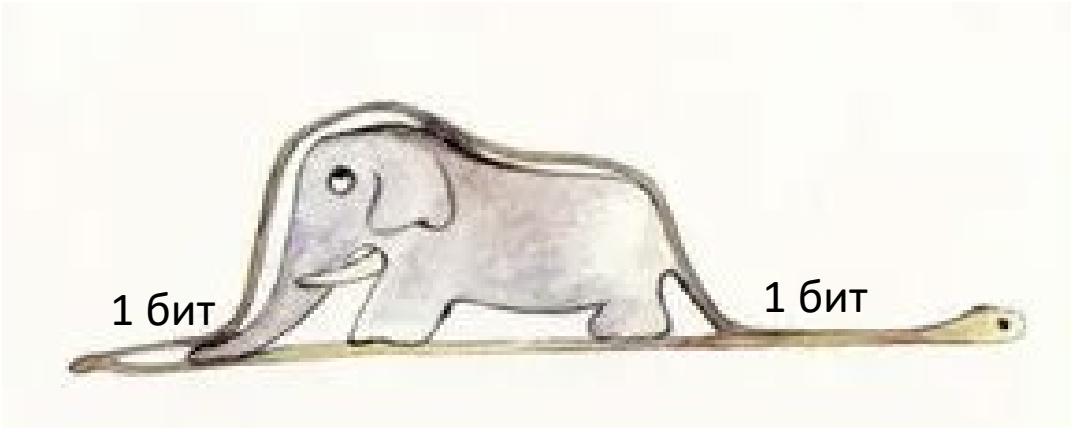
$$\nabla L(w) = \frac{1}{n}\sum_i \nabla L_i\,(w) \to E\nabla L_i(w), n \to \infty$$

Требуется непрерывность $E\nabla L_i(w)$ а не $\nabla L(w)$

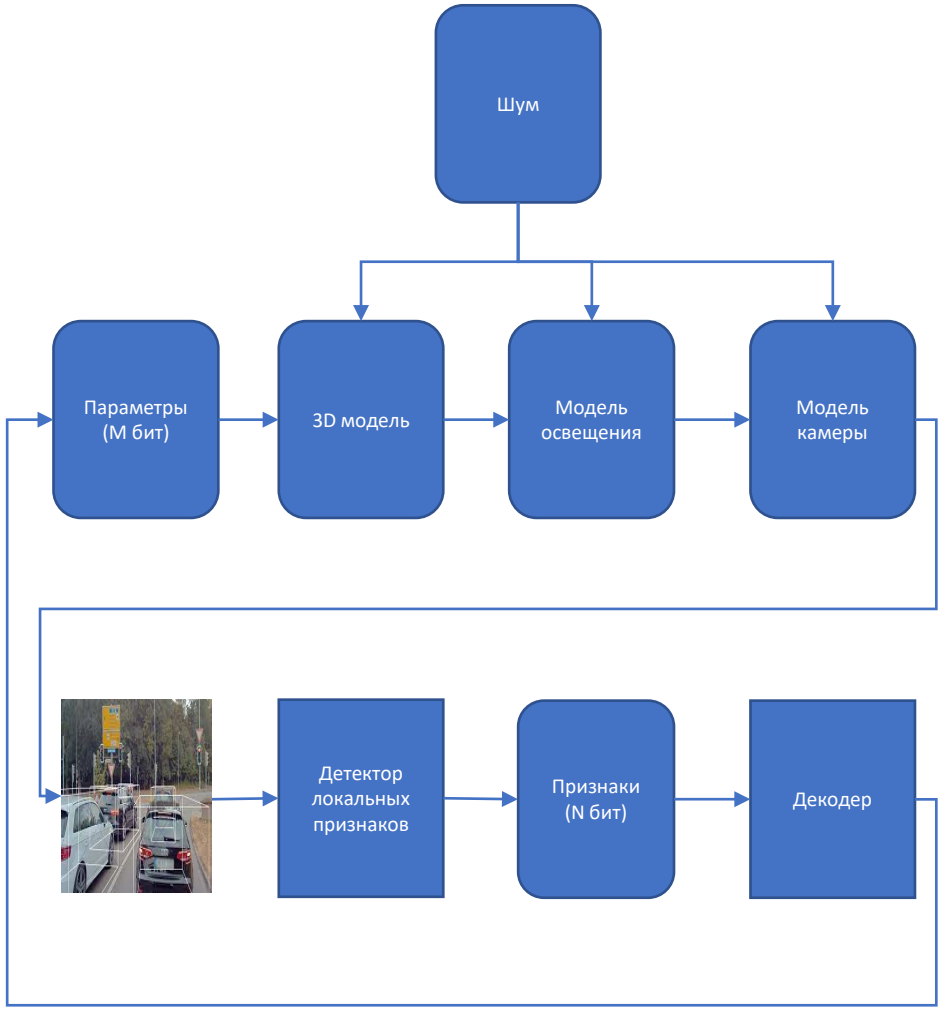
Каковы условия?

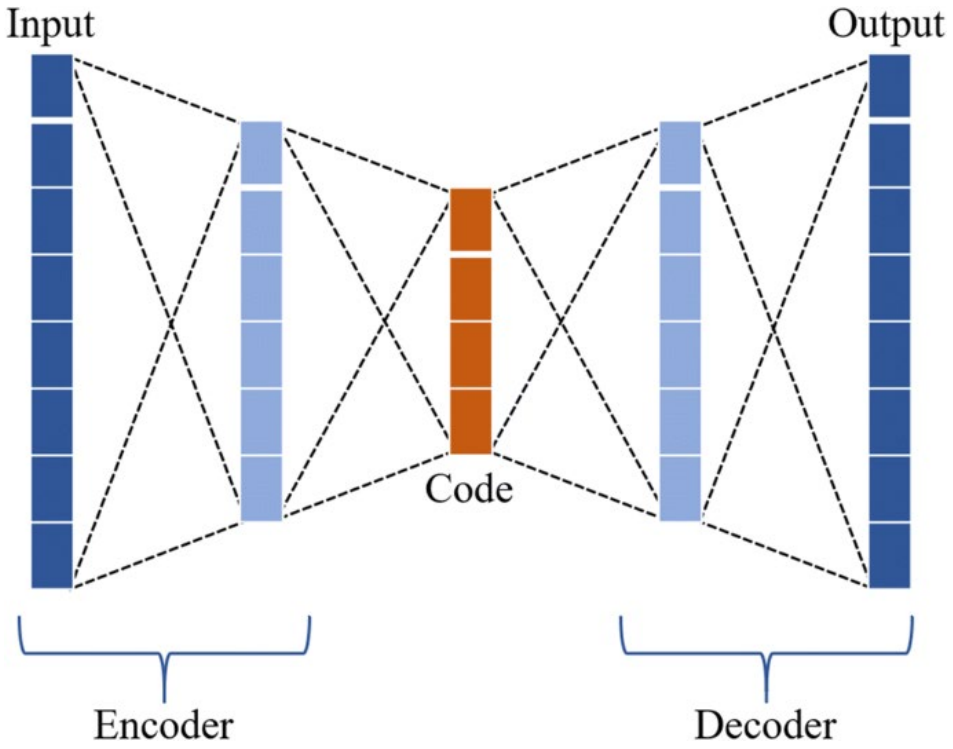
Бинаризация нейросетей. Шляпа

- Сожмем вход и выход в 1 бит
- Определим минимальное квантование внутренних слоев
- Расщепим слои и веса до достижения бинаризации



Метафора декодирования





Оптимальный код. Бустинг

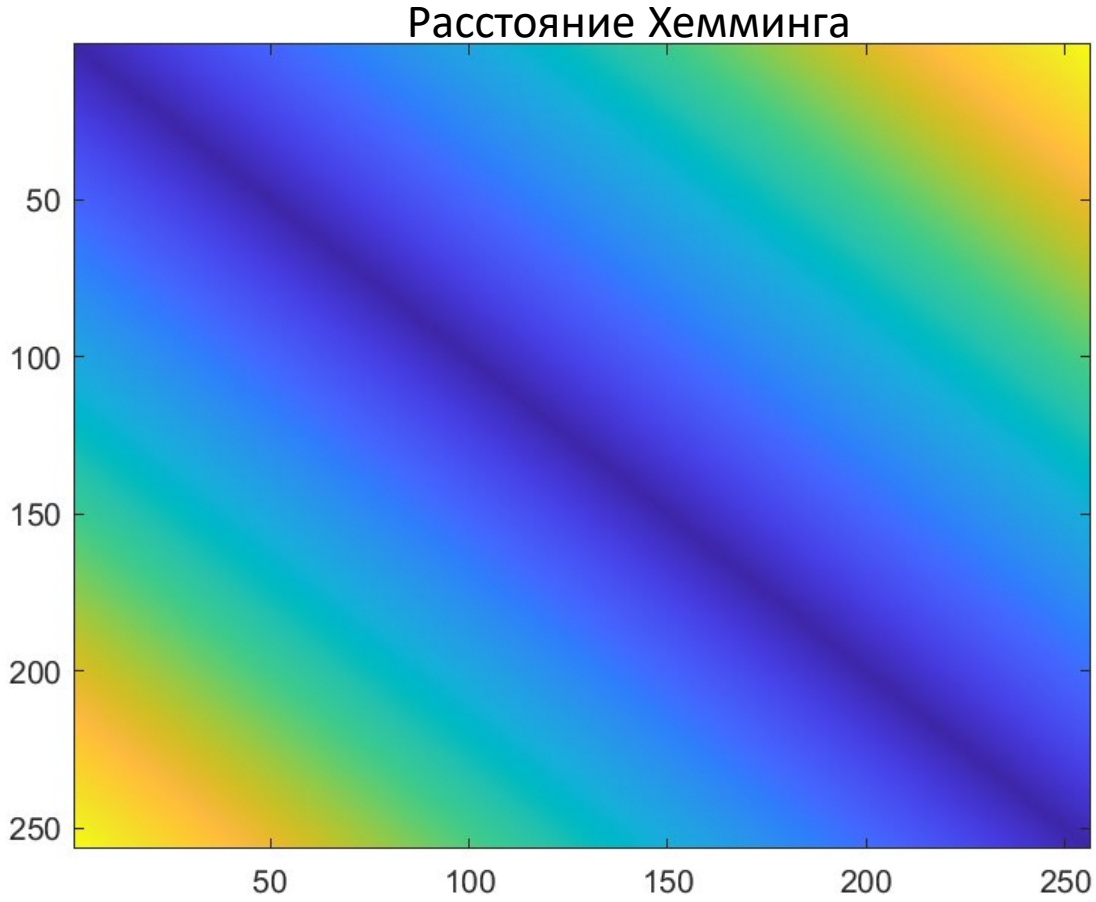
- Классы неупорядоченно
- Выход бинарный классификаторов – код коррекции ошибок
  - Расстояние Хемминга для декодирования
- Каждый бит это слабый классификатор
- Унитарный (One-hot) код – бинарный классификатор один ко многим
  - Несбалансированный
  - Экспоненциальный рост с ростом числа классов
- Можно ли лучше?
  - Бустинг
  - Нейросети = деревья решений
  - Бустинг → Нейробустинг = обучение нейросетевого полносвязного декодера
- Размер скрытого состояния
  - Теоремы Шенона для канала с шумами?

Полный код

- Самый лучший код – полный код
  - Все возможные различные двоичные классификаторы
- Экспоненциальный рост
  - 10 классов → 510 разбиений

Регрессия? Код

- Если классы упорядочены?
- Непозиционный двоичный код
$$c = 2^n - 1$$
- Укоротить?
  - Локальное Хеширование (Locality Sensitive Hashing), декодирование с помощью взвешенного расстояния Хемминга
  - Позиционный троичный код?



Цель

Нормализованная квантованная (в фикс точке) сеть

$$x_{i+1} = \min\left(1, ReLU\left(\frac{w_i x_i + w_{i,0} \cdot 1}{q_i}\right)\right) \ , x_i \in [0,1]^{k_i}, i \in [0..m-1]$$

$$w_{i,0} = -\mathrm{E}(w_i x_i)$$

$$q_i = quantile_p\left(w_i x_i + w_{i,0} \cdot 1\right), \Pr\left[w_i x_i + w_{i,0} \cdot 1 \ < \ q_i\right] = p$$

Цель – квантовать все $x_i \in \{0,1\}, w_i \in \{0,1\}$.

Греем

- Гипотеза I

Для каждого x_i, w_i можно конструктивно определить ширину $b_i, \bar{b}_i$ в битах, при которой равномерное квантование не имеет потерь.

Преобразование сети

Квантовать вход и выход сети, $x_i = y_i + s_i n_i$, s_i – скалярная амплитуда шума, y_i – непрерывны, $s_i n_i$ – шум квантования активаций

$$n_i = IID{\sim}U\left([-0.5,0.5]^{k_i}\right)$$

Квантовать веса сети, $w_i = v_i + \bar{s}_i \bar{n}_i$, $\bar{s}_i$– скалярная амплитуда шума, v_i – непрерывны, $\bar{s}_i \bar{n}_i$ – шум квантования весов

$$\bar{n}_i = IID{\sim}U\left([-0.5,0.5]^{(k_i+1)\times k_{i+1}}\right)$$

Заменим x_i на y_i и w_i на v_i

$$y_{i+1} = \min\left(1, ReLU\left(\frac{(v_i+\bar{s}_i\bar{n}_i)(y_i + s_i n_i) + v_{i,0}\cdot 1 + \bar{s}_i\bar{n}_{i,0}}{q_i}\right)\right) - s_{i+1}n_{i+1}$$

$$y_{i+1} = \min\left(1, ReLU\left(\frac{v_i y_i + v_{i,0}\cdot 1 + \bar{s}_i\bar{n}_{i,0} + \bar{s}_i\bar{n}_i y_i + s_i v_i n_i + s_i\bar{s}_i\bar{n}_i n_i}{q_i}\right)\right) - s_{i+1}n_{i+1}$$

$$y_i \in \left[-\frac{s_i}{2}, 1+\frac{s_i}{2}\right]^{k_i}$$

Условная оптимизация (постановка)

Определим функцию потерь $L(\Theta, s, \bar{s})$, $\Theta = [v_i], s = [s_i], \bar{s} = [\bar{s}_i]$
Обучим сеть $[y_i]$ с функцией потерь $L(\Theta) = L(\Theta, s = 0, \bar{s} = 0)$, найдем

$$L_{min} = \min L(\Theta) = L(\Theta_0)$$

Рассмотрим задачу условной оптимизации

$$x_i = y_i + s_i n_i, n_i = IID \sim U\left([-0.5,0.5]^{k_i}\right)$$

$$w_i = v_i + \bar{s}_i \bar{n}_i, \bar{n}_i = IID \sim U\left([-0.5,0.5]^{(k_i+1)\times k_{i+1}}\right)$$

$$\bar{g}_i = \max |w_i|$$

$$L^*(s, \bar{s}) = -\sum \ln s_i + \sum \ln \bar{g}_i - \sum \ln \bar{s}_i \, , L(\Theta, s, \bar{s}) \leq L_{min} + \epsilon, s_i > 0, \bar{s}_i > 0$$

y_i, v_i – непрерывные, ϵ – допустимые потери при квантовании, $s_i, \bar{s}_i$ – параметры распределения вероятностей, аналогичный параметрам вариационного автоэнкодера.

$$b_i = -\lfloor \log_2 s_i \rfloor, \bar{b}_i = \lceil \log_2 \bar{g}_i \rceil - \lfloor \log_2 \bar{s}_i \rfloor$$

Условная оптимизация (решение)

- Численное решение задачи условной оптимизации барьерным методом

$$L^* = -\sum \ln s_i + \sum \ln \bar{g}_i - \sum \ln \bar{s}_i \, , L(\Theta, s, \bar{s}) \leq L_{min} + \epsilon, s_i > 0, \bar{s}_i > 0$$

Перейдем к задаче безусловной оптимизации при помощи барьерного метода, определим новую функцию потерь

$$\hat{L}(\Theta, s, \bar{s}) = -\sum \ln s_i + \sum \ln \bar{g}_i - \sum \ln \bar{s}_i - \alpha \ln\left(L_{min} + \epsilon - L(\Theta, s, \bar{s})\right)$$

- Численное решение задачи условной оптимизации методом потенциалов

$$L^* = -\sum \ln s_i + \sum \ln \bar{g}_i - \sum \ln \bar{s}_i \, , L(\Theta, s, \bar{s}) \leq L_{min} + \epsilon, s_i > 0, \bar{s}_i > 0$$

Перейдем к задаче безусловной оптимизации при помощи метода потенциалов, определим новую функцию потерь

$$\hat{L}\left(\Theta, s, \bar{s}\right) = -\sum \ln s_i + \sum \ln \bar{g_i} - \sum \ln \bar{s_i} + \alpha\left(\max\left(0, L\left(\Theta, s, \bar{s}\right) - L_{min} - \epsilon\right)\right)^{\beta}$$

Расщепляем

- Гипотеза II

Для слоя с $b_i \geq 2$ можно, изменив архитектуру сети, перейти к $\hat{b}_i = b_i - 1$

Преобразование сети

Расщепим $x_i = 0.5 - \frac{x_{i+1}^-}{2} + \frac{x_{i+1}^+}{2}$; $x_{i+1}^-, x_{i+1}^+ \in [0,1]$

$x_{i+1}^+ = ReLU(2x_i - 1), x_{i+1}^- = ReLU(1 - 2x_i)$ на один бит уже x_i

Вставим новый слой $\tilde{x}_{i+2} = x_{i+1} = \min\left(1, ReLU\left(q_i^{-1}\left(w_i\left(\frac{x_{i+1}^+}{2} - \frac{x_{i+1}^-}{2}\right) + \frac{w_i}{2} - \mu_i\right)\right)\right)$

- Гипотеза III

Для слоя с $\bar{b}_i \geq 2$ можно, изменив архитектуру сети, перейти к $\bar{\bar{b}}_i = \bar{b}_i - 1$

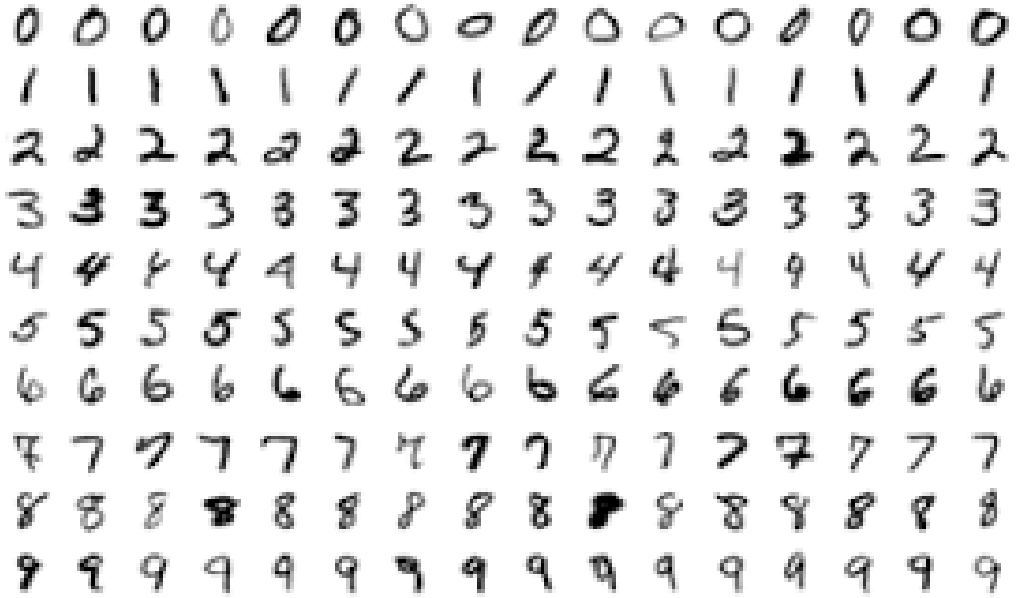
Преобразование сети

Расщепим $w_i = w_i^+ + w_i^-$,

$w_i^+ = \left\lceil \frac{w_i}{s_i^*} \right\rceil, w_i^- = \left\lfloor \frac{w_i}{s_i^*} \right\rfloor$, на один бит уже

MNIST

- Рукописные цифры 28x28x8 бит
- 10 классов по 1000 картинок в тренировочных данных



- Структура

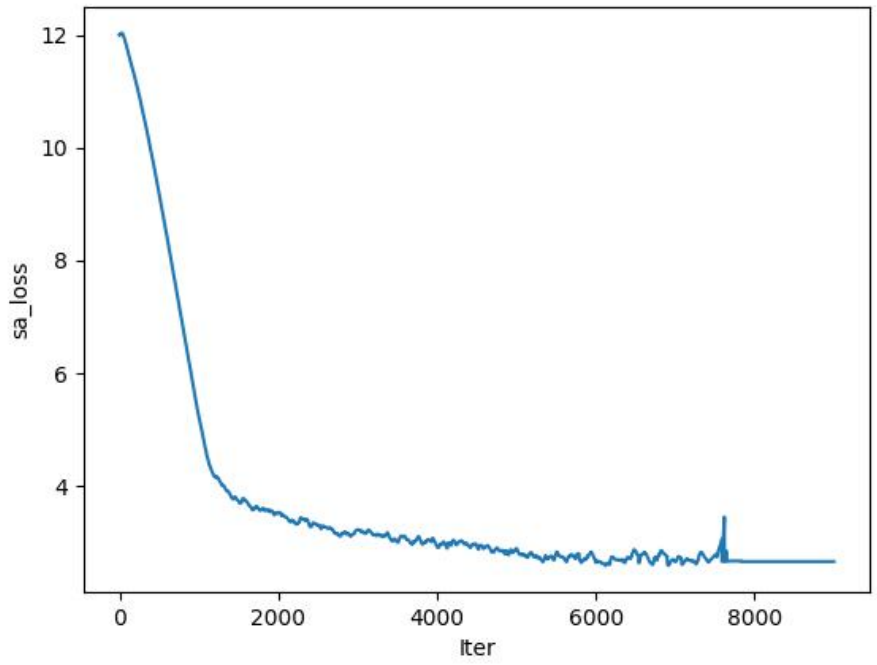
  conv_block(4, 8, 5)
  conv_block(8, 16, 3)
  conv_block(16, 32, 3)
  conv_block(32, 64, 3)
  self._fc(64)

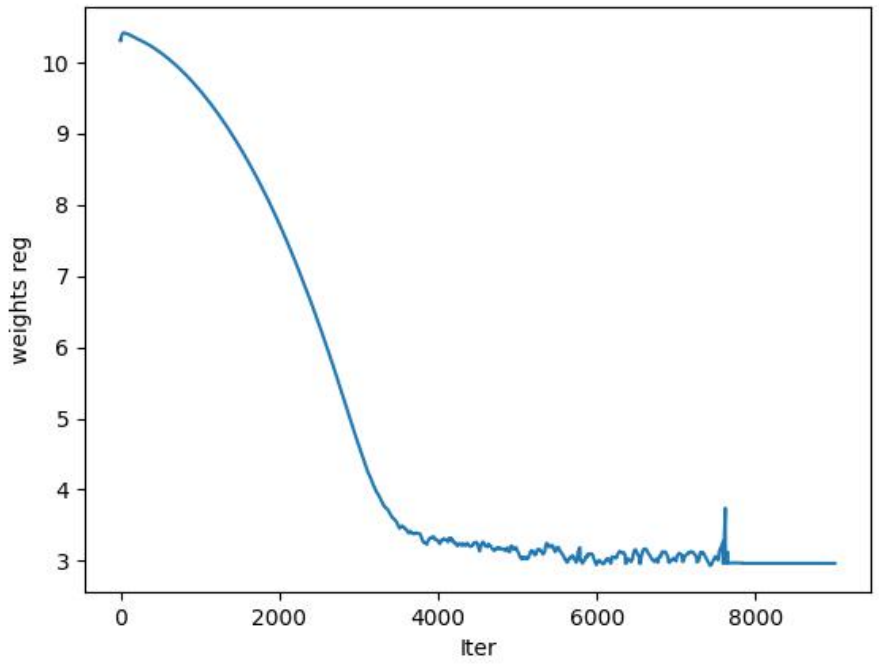
- 4 сверточных слоя conv+relu+maxpol
- 1 полносвязный слой
- Бинарные коды на входе и выходе
- Функция потерь – BinaryCrossEntropy

Постановка эксперимента

1. Переобучаем сеть в плавающей точке до 100% точности на тренировочных данных
2. Фиксируем значение функции потерь
3. Решаем задачу условной оптимизации с сохранением значения функции потерь методом потенциалов $\alpha = \frac{1}{L^*},\ \beta = 1$
4. Валидируем 100% точности в фиксированной точке

Результаты квантования (расщепленная по весам сеть)





Расщепление весов

- Расщепленная сеть

  Model weights bit_width

  1.init_conv.conv2d.weight: 4.0

  1.conv_block_1.conv2d.weight: 4.0

  1.conv_block_2.conv2d.weight: 4.0

  1.conv_block_3.conv2d.weight: 4.0

  1.conv_block_4.conv2d.weight: 4.0

  1.fc.lin.weight: 4.0

- Validating activations 100%

- Базовая сеть

  Model weights bit_width

  1.init_conv.conv2d.weight: 6.0

  1.conv_block_1.conv2d.weight: 7.0

  1.conv_block_2.conv2d.weight: 7.0

  1.conv_block_3.conv2d.weight: 7.0

  1.conv_block_4.conv2d.weight: 6.0

  1.fc.lin.weight: 5.0

- Validating activations 100%

Выводы

- Все очень интересно
- Квантование нагреванием работает
- Проблемы со сходимостью условной оптимизации
- Увеличение ширины сети приводит к понижению разрядности
- Следующие шаги
  - Сеть больших размеров (CIFAR 10, ResNet20)
  - Реализация более эффективных эмбедингов
  - Автоматическое расщепление