BatchNorm и LayerNorm

Дима Авдеев (ВШЭ), Игнат Чернышов (МФТИ), Влад Кузнецов (ментор)

AI360, Москва

Постановка проблемы

- Хотим обучить глубокую нейроную сеть.
- Для глубоких нейронных сетей измение одного слоя может сильно изменить следующий
- Возникает проблема долгого обучения, тщательного подбора гиперпараметров.
- Для ускорения обучения используется разбиение train выборки на батчи.
- Возникает идея посмотреть на матожиданние и отклонение данных.

Идеи BatchNorm и LayerNorm

- Статьи [1, 2] предлагают нормализацию данных после каждого слоя, предлагают менять среднее отклонение(отвечает за дисперсию) и среднее(отвечает за матожидание) данных.
- Batch предлагает брать квадратное среднее отклонение и среднее по батчу, Layer — по слою.
- Для каждого параметра/координаты x_i считаем $\hat{x_i}$, которое считается по батчу $\mathcal{B}$ через среднее батча $\mu_{\mathcal{B}}$ и среднее квадратное отклонение $\sigma^2_{\mathcal{B}}$:

$$\mu_{\mathcal{B}} = \frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|} x_i$$

$$\sigma^2_{\mathcal{B}} = \frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|} (x_i - \mu_{\mathcal{B}})^2$$

$$\hat{x}_i = \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \varepsilon}}$$

- ε выбирается любым маленьким, он нужен лишь для избежания деления на 0.
- Далее для каждого перцептрона обучается следующие параметры.

$$y_i = \gamma\hat{x_i} + \beta$$

- После окончания обучения параметры «замораживаются», после чего делают дополнительные вычисления для избежания смещённой оценки. Мы опускаем обозначение индекса нейрона (k).

$$E[x] \leftarrow E_{\mathcal{B}}[\mu_{\mathcal{B}}],\ Var[x] \leftarrow \frac{m}{m-1}E_{\mathcal{B}}[\sigma^2_{\mathcal{B}}]$$

$$y = \frac{\gamma}{\sqrt{Var[x] + \varepsilon}} \cdot x + (\beta - \frac{\gamma \cdot E[x]}{\sqrt{Var[x] + \varepsilon}})$$

- Доказываются некоторые полезные свойства нормализации, например при умножении на скаляр сохраняются свойства матрицы Якоби.

$$BN(Wu + b) = BN(Wu) = BN((aW)u)$$

$$\frac{\partial BN((aW)u)}{\partial u} = \frac{\partial BN(Wu)}{\partial u}$$

$$\frac{\partial BN((aW)u)}{\partial (aW)} = \frac{1}{a} \cdot \frac{\partial BN(Wu)}{\partial W}$$

Эксперименты

В оригинальной статье [1] проводились эксперименты с Batch нормализацией и без неё по обучению многослойного полносвязного перцептрона на MNIST(определение числа по картинке 28×28 пикселей) со следующими параметрами:

- 3 полносвязных скрытых слоя по 100 нейронов каждый.
- Функция активации: сигмоида. Используется стандартный градиентный спуск для обучения.

Мы повторили данный эксперимент, кроме того, была добавленна Layer нормализация для того чтобы сравнить методы из статьей [1, 2]. В таблице ниже приведенны вычисления accuracy после каждых 5000 шагов.

Таблица 1: Сравнение итоговой точности на разных шагах во времени обучения MNIST

Model	Acc1, %	Acc2, %	Acc3, %	Acc4, %	Acc5, %	Acc6, %	Acc7, %	Acc8, %	Acc9, %	Acc10, %
Layer Norm	95.2	95.9	97.2	97.5	97.1	97.3	97.7	97.8	97.3	97.8
Batch Norm	96.2	97.2	97.6	97.9	97.9	98.0	98.0	97.9	98.2	98.2
Without Norm	79.2	92.0	94.9	95.9	96.6	96.6	97.2	97.3	97.2	97.1

Ниже приведенны графики accuracy и cross entropy loss, полученные на этих трёх моделях на MNIST. Далее мы взяли Fashion MNIST и попытались сделать максимальное качество модели. На Fashion MNIST были выбраны другие гиперпараметры для повышения accuracy, кроме того авторы не делали экспериментов на этом датасете, ниже приведены графики.

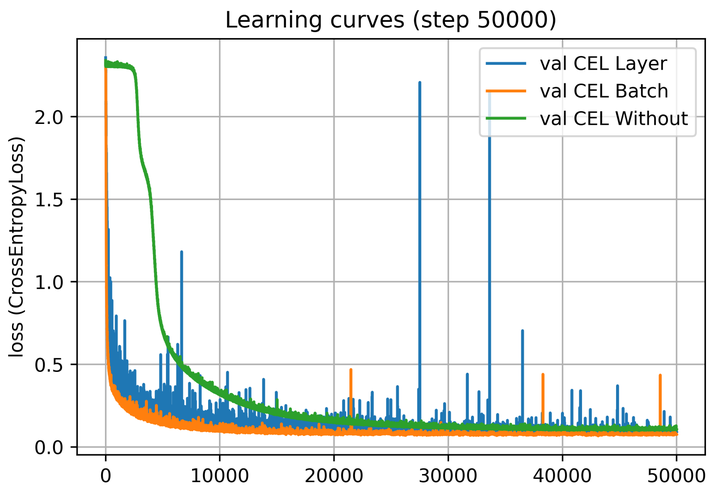

Рис. 1: Cross Entropy Loss на MNIST

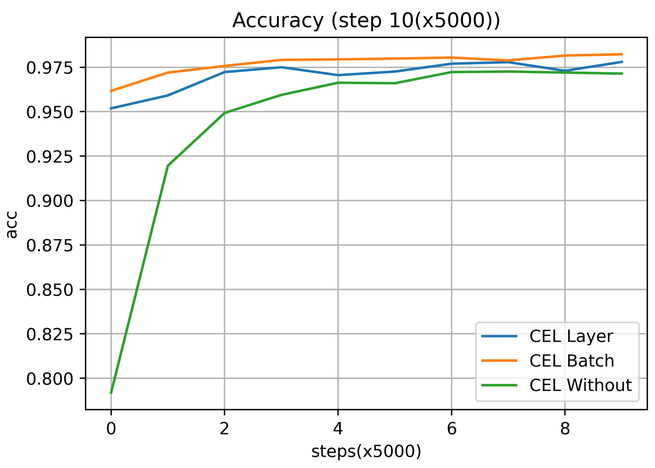

Рис. 2: Метрика Accuracy на MNIST

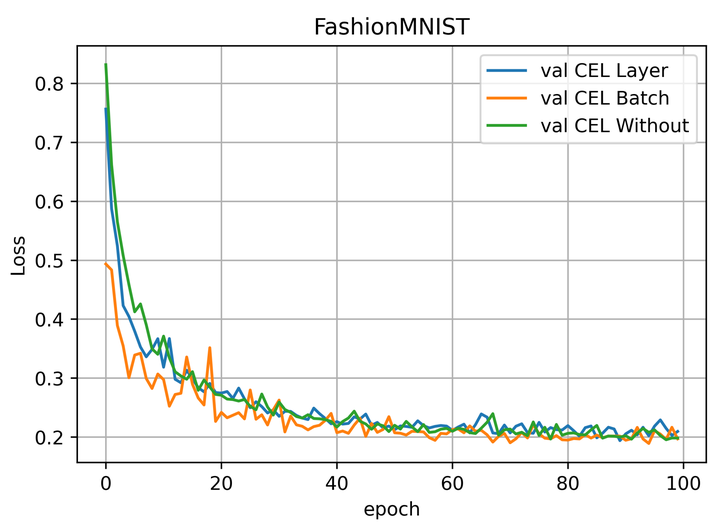

Рис. 3: Cross Entropy Loss на Fashion-MNIST

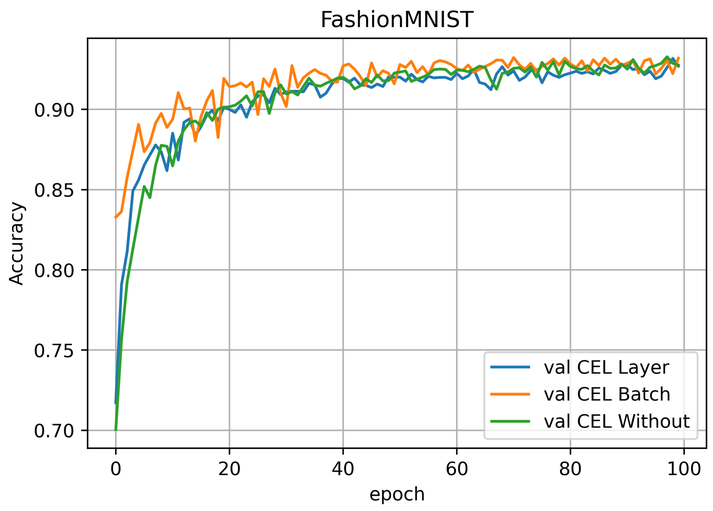

Рис. 4: Метрика Accuracy на Fashion-MNIST

Выводы

В результате работы получилось повторить опыты оригинальной статьи [1] с дополнительной Layer нормализацией и было выясненно, что и Batch и Layer нормализации примерно одинаково быстро сходятся, в отличие от модели без нормализации. Опыты показывают, что Layer похож по качеству accuracy Batch нормализации, обе нормализации улучшают качество.

Список литературы

[1] Ioffe S., Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift //International conference on machine learning. – pmlr, 2015. – С. 448-456.

[2] Ba J. L., Kiros J. R., Hinton G. E. Layer normalization //arXiv preprint arXiv:1607.06450. – 2016.