Глава 1

Полносвязные Сети

- $\mathcal{N}$ — размер батча
- $\mathcal{D}$ — размерность вектора признаков

1.1 Batch Normalization

Пусть $\boldsymbol{x}$ — это вектор на входе слоя ($x \in \mathbb{R}^{\mathcal{N}}$). Тогда вектор $\boldsymbol{y}$ на выходе слоя есть

$$\boldsymbol{y} = \gamma \cdot \frac{\boldsymbol{x} - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \boldsymbol{\beta},$$

где

$$\mu = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} x_i, \qquad \sigma^2 = \frac{1}{\mathcal{N}} \sum_{i=1}^{\mathcal{N}} (x_i - \mu)^2 = \frac{1}{\mathcal{N}} x_i^2 - \left(\frac{1}{\mathcal{N}} x_i \right)^2.$$

Пусть дана производная функции потерь по $\boldsymbol{y}$, т.е. $\partial \mathcal{L} / \partial \boldsymbol{y}$. Найдем производные функции потерь по $\boldsymbol{x}$, $\boldsymbol{\beta}$, γ. Сначала найдем $\partial \mathcal{L} / \partial \boldsymbol{x}$:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{x}} = \underbrace{\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}}_{\mathbb{R}^{\mathcal{N} \times \mathcal{N}}} \cdot \underbrace{\frac{\mathcal{L}}{\boldsymbol{y}}}_{\mathbb{R}^{\mathcal{N}}}, \tag{1.1}$$

где

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} = \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_2}{\partial x_1} & \cdots & \frac{\partial y_{\mathcal{N}}}{\partial x_1} \\ \frac{\partial y_1}{\partial x_2} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_{\mathcal{N}}}{\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_1}{\partial x_{\mathcal{N}}} & \frac{\partial y_2}{\partial x_{\mathcal{N}}} & \cdots & \frac{\partial y_{\mathcal{N}}}{\partial x_{\mathcal{N}}} \end{pmatrix} \tag{1.2}$$

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} = \gamma \frac{\partial \left(\frac{\boldsymbol{x} - \mu}{\sqrt{\sigma^2 + \varepsilon}} \right)}{\partial \boldsymbol{x}}.$$

Рассмотрим производную

$$\frac{\partial y_i}{\partial x_j} = \gamma \partial \left(\frac{x_i - \mu}{\sqrt{\sigma^2 + \varepsilon}} \right) / \partial x_j = \gamma \frac{\partial (x_i - \mu)}{\partial x_j} \frac{1}{\sqrt{\sigma^2 + \varepsilon}} - \frac{\gamma}{2} \frac{x_i - \mu}{(\sigma^2 + \varepsilon)^{3/2}} \frac{\partial (\sigma^2 + \varepsilon)}{\partial x_j} =$$
$$= \gamma \left(\delta_{ij} - \frac{1}{\mathcal{N}} \right) \frac{1}{\sqrt{\sigma^2 + \varepsilon}} - \frac{\gamma}{2} \frac{x_i - \mu}{(\sigma^2 + \varepsilon)^{3/2}} \cdot 2 \left(\frac{x_j}{\mathcal{N}} - \frac{\mu}{\mathcal{N}} \right) = \gamma \left(\delta_{ij} - \frac{1}{\mathcal{N}} \right) \frac{1}{\sqrt{\sigma^2 + \varepsilon}} - \gamma \frac{(x_i - \mu)(x_j - \mu)}{\mathcal{N} (\sigma^2 + \varepsilon)^{3/2}}.$$

Таким образом

$$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} = \frac{\gamma}{\sqrt{\sigma^2 + \varepsilon}} \left(\left(\boldsymbol{I} - \frac{1}{\mathcal{N}} \boldsymbol{E} \right) - \frac{\boldsymbol{C}}{\sigma^2 + \varepsilon} \right),$$

где $\boldsymbol{C} = \frac{1}{\mathcal{N}}(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^T$, $\boldsymbol{I}$ — единичная матрица размера $\mathcal{N}\times\mathcal{N}$, $\boldsymbol{E}$ — матрица, полностью состящая из единиц, размера $\mathcal{N}\times\mathcal{N}$. Тогда

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{x}} = \frac{\gamma}{\sqrt{\sigma^2+\varepsilon}}\left(\left(\boldsymbol{I}-\frac{1}{\mathcal{N}}\boldsymbol{E}\right)-\frac{\boldsymbol{C}}{\sigma^2+\varepsilon}\right)\frac{\partial\mathcal{L}}{\partial\boldsymbol{y}} = \frac{\gamma}{\sqrt{\sigma^2+\varepsilon}}\left(\frac{\partial\mathcal{L}}{\partial\boldsymbol{y}}-\boldsymbol{e}\cdot\overline{\frac{\partial\mathcal{L}}{\partial\boldsymbol{y}}}-\frac{\boldsymbol{x}-\mu}{\mathcal{N}(\sigma^2+\varepsilon)}\cdot\langle\boldsymbol{x}-\mu,\frac{\mathcal{L}}{\boldsymbol{y}}\rangle\right),$$

где $\boldsymbol{e}$ — столбец из единиц размерности $\mathcal{N}$, $\overline{\frac{\partial\mathcal{L}}{\partial\boldsymbol{y}}}$ — среднее значение элементов вектора $\frac{\partial\mathcal{L}}{\partial\boldsymbol{y}}$. Обозначив через $\boldsymbol{z}$ “стандартизованный” вектор $\boldsymbol{x}$ ($\boldsymbol{z}=(\boldsymbol{x}-\mu)/\sigma$), получим

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{x}} = \frac{\gamma}{\sqrt{\sigma^2+\varepsilon}}\left(\frac{\partial\mathcal{L}}{\partial\boldsymbol{y}}-\boldsymbol{e}\cdot\overline{\frac{\partial\mathcal{L}}{\partial\boldsymbol{y}}}-\frac{\boldsymbol{x}-\mu}{\mathcal{N}(\sigma^2+\varepsilon)}\cdot\langle\boldsymbol{x}-\mu,\frac{\mathcal{L}}{\boldsymbol{y}}\rangle\right) = \frac{\gamma}{\sqrt{\sigma^2+\varepsilon}}\left(\frac{\partial\mathcal{L}}{\partial\boldsymbol{y}}-\boldsymbol{e}\cdot\overline{\frac{\partial\mathcal{L}}{\partial\boldsymbol{y}}}-\frac{\boldsymbol{z}}{\mathcal{N}}\cdot\langle\boldsymbol{z},\frac{\mathcal{L}}{\boldsymbol{y}}\rangle\right).$$

Теперь найдем производные $\partial\mathcal{L}/\partial\boldsymbol{\beta}$ и $\partial\mathcal{L}/\partial\gamma$:

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\beta}} = \frac{\partial\boldsymbol{y}}{\partial\boldsymbol{\beta}}\frac{\partial\mathcal{L}}{\partial\boldsymbol{y}} = \boldsymbol{I}\frac{\partial\mathcal{L}}{\partial\boldsymbol{y}} = \frac{\partial\mathcal{L}}{\partial\boldsymbol{y}},$$
$$\frac{\partial\mathcal{L}}{\partial\gamma} = \left(\frac{\partial\boldsymbol{y}}{\partial\gamma}\right)^T\frac{\partial\mathcal{L}}{\partial\boldsymbol{y}} = \frac{\boldsymbol{x}-\mu}{\sqrt{\sigma+\varepsilon}}$$

1.1.1 Реализация в Python

Представленная ниже реализация предполагает, что

- $\boldsymbol{X}\in\mathbb{R}^{\mathcal{N}\times\mathcal{D}}$ — матрица объектов-признаков
- $\boldsymbol{Z}\in\mathbb{R}^{\mathcal{N}\times\mathcal{D}}$ — “стандартизованная” вдоль оси 0 матрица $\boldsymbol{X}$, т.е. выход слоя `BatchNormalization`
- $\boldsymbol{\mu}\in\mathbb{R}^{\mathcal{N}}$ — средние значения для признаков
- $\boldsymbol{\sigma}\in\mathbb{R}^{\mathcal{N}}$ — стандартные отклонения для признаков

```
def batchnorm_backward_alt(output_grad, cache):
  X, Z, mu, sigma, gamma, beta, eps = cache
  N, D = X.shape
  X_grad = gamma / np.sqrt(sample_var + eps)[None, :] *\
          ((output_grad - np.mean(output_grad, axis=0)[None, :]) -\
          Z * np.mean(np.multiply(Z, output_grad), axis=0)[None, :])
  beta_grad = np.sum(output_grad, axis=0)
  gamma_grad = np.sum(np.multiply(X_norm, output_grad), axis=0)
  return X_grad, gamma_grad, beta_grad
```