Feed-Forward Neural Network

Forward Propagation

Single Data Point

$$m = W^{(1)}x + b^{(1)}$$
$$h = \mathrm{ReLU}(m)$$
$$z = W^{(2)}h + b^{(2)}$$
$$y = \mathrm{softmax}(z)$$
$$\mathcal{L} = \mathcal{L}_{CE}(y, t)$$

Batch

$$M = X(W^{(1)})^T + \mathbb{1}(b^{(1)})^T$$
$$H = \mathrm{ReLU}(M)$$
$$Z = H(W^{(2)})^T + \mathbb{1}(b^{(2)})^T$$
$$Y = \mathrm{softmax}(Z)$$
$$\mathcal{E} = \frac{1}{N}\sum_i \mathcal{L}_{CE}(y^{(i)}, t^{(i)})$$

Backpropagation

$$\overline{\mathcal{L}} = 1$$
$$\bar{z} = y - t$$
$$\bar{h} = (W^{(2)})^T \bar{z}$$
$$\overline{W^{(2)}} = \bar{z}(h)^T$$
$$\overline{b^{(2)}} = \bar{z}$$
$$\bar{m} = \bar{h} \circ \mathrm{ReLU}'(m)$$
$$\overline{W^{(1)}} = \bar{m}(x)^T$$
$$\overline{b^{(1)}} = \bar{m}$$

$$\overline{\mathcal{L}} = 1$$
$$\bar{Z} = \frac{1}{N}(Y - T)$$
$$\bar{H} = \bar{Z}W^{(2)}$$
$$\overline{W^{(2)}} = (\bar{Z})^T H$$
$$\overline{b^{(2)}} = (\bar{Z})^T \mathbb{1}$$
$$\bar{M} = \bar{H} \circ \mathrm{ReLU}'(M)$$
$$\overline{W^{(1)}} = (\bar{M})^T X$$
$$\overline{b^{(1)}} = (\bar{M})^T \mathbb{1}$$

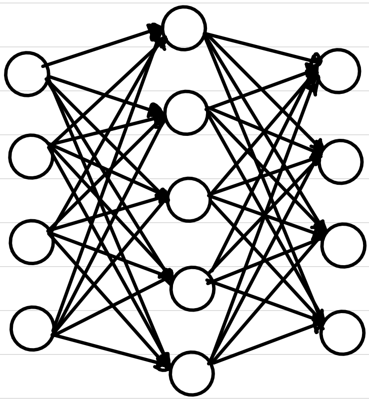
Adam (Adaptive Moment Estimation)

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\frac{\partial \mathcal{L}}{\partial w_t}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)\left(\frac{\partial \mathcal{L}}{\partial w_t}\right)^2$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \qquad \hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

$$w_{t+1} = w_t + \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + e}$$



Gradient Clipping

$$g = \eta \frac{g}{\|g\|}$$

Weight Initialization

Xavier Initialization

$$x = \sqrt{\frac{6}{n_{inputs} + n_{outputs}}}$$

sigmoid or softmax

$$W \sim \mathrm{Uniform}(-x, x)$$

He / Kaiming Initialization

$$W \sim \mathrm{Uniform}\left(-\sqrt{\frac{6}{n_{inputs}}}, \sqrt{\frac{6}{n_{outputs}}}\right)$$

ReLU

Dropout

$$h' = \begin{cases} 0 & \text{with probability } p \\ \frac{h}{1-p} & \text{otherwise} \end{cases}$$

Note:

* for code get rid of $\mathbb{1}$ and transpose → let numpy work