LightNet - Supplementary Materials

Chengxi Ye, Chen Zhao, Yezhou Yang
Cornelia Fermüller and Yiannis Aloimonos

February 26, 2017

1 Supplementary Information

1.1 Layers

1.1.1 Notations

z: the final layer output.
x: the input of a layer.
y: the output of a layer.
$y = g(x)$.
$z = f(y)$.
$\frac{\partial z}{\partial x}$: given column vector x, the Jacobian is a row vector $[\frac{\partial z}{\partial x_1}, \frac{\partial z}{\partial x_2}, ..., \frac{\partial z}{\partial x_n}]$.
$\frac{\partial z}{\partial W}$: given matrix W_{ij}, the Jacobian is a matrix $(\frac{\partial z}{\partial W_{ij}})^T$ (Note there is a transpose here).

1.1.2 Linear Layer

The forward transform is:

$$y = Wx + b \tag{1}$$

The backward pass:

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y}\frac{\partial y}{\partial x} = \frac{\partial z}{\partial y}W \tag{2}$$

$$\frac{\partial z}{\partial b} = \frac{\partial z}{\partial y}\frac{\partial y}{\partial b} = \frac{\partial z}{\partial y} \tag{3}$$

Next we have $y^T = x^T W^T + b^T$, therefore:

$$\frac{\partial z}{\partial W^T} = \frac{\partial z}{\partial y^T}\frac{\partial y^T}{\partial W^T} = \frac{\partial z}{\partial y^T}x^T \tag{4}$$

1.1.3 ReLU Layer

The forward transform is:

$$y = Id(x > 0)x \tag{5}$$

The backward pass:

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y}\frac{\partial y}{\partial x} = \frac{\partial z}{\partial y}Id(x > 0) \tag{6}$$

1.1.4 Softmax Layer

The softmax function maps the input x to the output probability y.

The forward transform is:

$$y_i = \frac{e^{x_i}}{\sum_{j=1}^{C} e^{x_j}} \tag{7}$$

The backward pass: Let $S = \sum_{i=1}^{C} e^{x_i}$,

$$\frac{\partial y_i}{\partial x_j} = \frac{\delta_{ij} e^{x_i} S - e^{x_i} e^{x_j}}{S^2} = \delta_{ij} y_i - y_i y_j \tag{8}$$

Therefore,

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y}\frac{\partial y}{\partial x} = \frac{\partial z}{\partial y} diag(y) - \frac{\partial z}{\partial y} y y^T \tag{9}$$

1.1.5 Cross Entropy/Softmax Log loss Layer

The loss of a classification can be defined as the negative log likelihood. If softmax is used to calculate the probability, we have $-log(p_i)$ as the loss of a given sample, where i is the ground truth category. Let $S = \sum_{j=1}^{C} e^{x_j}$, The forward transform is:

$$z = -log \frac{e^{x_i}}{\sum_{j=1}^{C} e^{x_j}} = log(\sum_{j=1}^{C} e^{x_j}) - x_i = log(S) - x_i \tag{10}$$

The backward pass:

$$\frac{\partial z}{\partial p_i} = -\frac{1}{p_i}([\delta_{ij} p_i] - p_i p^T) = -\frac{1}{p_i}([0, ..., p_i, ..., 0] - p_i [p_1, ..., p_C]) = [p_1, ..., p_C] - [0, ... 1_i, ..., 0] \tag{11}$$

1.1.6 Batch Normalization Layer (To be rewritten)

The following formulas are used to implement the batch normalization layer.

$\mu_t = \frac{1}{b}\sum_{i=1}^{b} x_i (1-m) + m\mu_{t-1}$

$\sigma_t^2 = \frac{1}{b}\sum_{i=1}^{b}(x_i - \mu_t)^2 (1-m) + m\sigma_{t-1}^2$

$\hat{x_i} = \frac{x_i - \mu_t}{\sqrt{\sigma_t^2 + \epsilon}}$

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

$\frac{\partial z}{\partial \gamma} = \sum_i \frac{\partial z}{\partial y_i} \cdot \hat{x_i}$

$\frac{\partial z}{\partial \beta} = \sum_i \frac{\partial z}{\partial y_i}$

$\frac{\partial y_j}{\partial x_i} = \frac{\gamma}{\sqrt{\sigma_t^2 + \epsilon}}(\delta ij - \frac{1-m}{b}) - \frac{\gamma}{b} \cdot \frac{(x_j - \mu_t)}{(\sigma_t^2 + \epsilon)^{\frac{3}{2}}} \cdot (x_i - \mu_t)$

$\frac{\partial z}{\partial x_i} = \sum_j \frac{\partial z}{\partial y_j}\frac{\partial y_j}{\partial x_i} = \frac{\gamma}{\sqrt{\sigma_t^p + \epsilon}}(\frac{\partial z}{\partial y_i} - \frac{1-m}{b}\frac{\partial z}{\partial \beta} - \frac{1-m}{b}\frac{\partial z}{\partial \gamma}\frac{(x_i - \mu_t)}{\sqrt{\sigma_t^2 + \epsilon}})$

1.2 A Recurrent Neural Network (with Skip Links)

The modules in LightNet can also be used to implement recurrent neural networks (RNN). We consider a simple RNN model:

$$\Delta h_t = f_h(W_h h_{t-1} + W_x x_t + b_h) \tag{12}$$

$$h_t = \lambda h_{t-1} + \Delta h_t \tag{13}$$

$$z_t = f_z(h_t) \tag{14}$$

$$z = \sum_{1 \le t \le T} (z_t) \tag{15}$$

In Eq. 13, $\lambda = 0, 1$ are important special cases. The equation reduces to the standard RNN formulation when $\lambda = 0$. We add skip links in Eq. 13 when $\lambda = 1$. z_t measures the loss in time frame t. The total loss is the sum of the loss in all time frames (Eq. 15).

In the back propagation process, one needs to use an important inductive property:

$$\frac{\partial z}{\partial h_T} = \frac{\partial z_T}{\partial h_T} \tag{16}$$

$$\frac{\partial z}{\partial h_{t-1}} = \frac{\partial z}{\partial h_t}\frac{\partial h_t}{\partial h_{t-1}} + \frac{\partial z_{t-1}}{\partial h_{t-1}} \tag{17}$$

$$\frac{\partial h_t}{\partial h_{t-1}} = \lambda + \frac{\partial \Delta h_t}{\partial h_{t-1}} \tag{18}$$

1.3 Gated Recurrent Unit (GRU)

The GRU architecture has two gates, one update gate:

$$U_t = sigmoid(W_{uh} h_{t-1} + W_{ux} x_t + b_u), \tag{19}$$

and the other reset gate:

$$R_t = sigmoid(W_{rh} h_{t-1} + W_{rx} x_t + b_r). \tag{20}$$

$$\Delta h_t = f_h(W_{hh}(R_t \odot h_{t-1}) + W_{hx} x_t + b_h) \tag{21}$$

$$h_t = (1 - U_t) \odot h_{t-1} + U_t \odot \Delta h_t \tag{22}$$

$$z_t = f_z(h_t) \tag{23}$$

$$z = \sum_{1 \le t \le T} (z_t) \tag{24}$$

In the back propagation process, one needs to use the inductive property:

$$\frac{\partial z}{\partial h_T} = \frac{\partial z_T}{\partial h_T} \tag{25}$$

$$\frac{\partial z}{\partial h_{t-1}} = \frac{\partial z}{\partial h_t}\frac{\partial h_t}{\partial h_{t-1}} + \frac{\partial z_{t-1}}{\partial h_{t-1}} \tag{26}$$

$$\frac{\partial h_t}{\partial h_{t-1}} = (1 - U_t) + (\Delta h_t - h_{t-1}) \odot \frac{\partial U_t}{\partial h_{t-1}} + U_t \odot \frac{\partial \Delta h_t}{\partial h_{t-1}} \tag{27}$$

1.4 LSTM Network

The forward process in an LSTM model can be formulated as:

$$i_t = sigmoid(W_{ih}h_{t-1} + W_{ix}x_t + b_i), \tag{28}$$

$$o_t = sigmoid(W_{oh}h_{t-1} + W_{ox}x_t + b_o), \tag{29}$$

$$f_t = sigmoid(W_{fh}h_{t-1} + W_{fx}x_t + b_f), \tag{30}$$

$$g_t = tanh(W_{gh}h_{t-1} + W_{gx}x_t + b_g), \tag{31}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t, \tag{32}$$

$$h_t = o_t \odot tanh(c_t), \tag{33}$$

$$z_t = f(h_t), z = \sum_{t=1}^{T} z_t. \tag{34}$$

Where $i_t/o_t/f_t$ denotes the response of the input/output/forget gate at time t. g_t denotes the distorted input to the memory cell at time t. c_t denotes the content of the memory cell at time t. h_t denotes the hidden node value. f maps the hidden nodes to the network loss z_t at time t. The full network loss is calculated by summing the loss at each individual time frame in Eq. 34.

To optimize the LSTM model, back propagation through time is implemented and the most critical value to calculate in LSTM is: $\frac{\partial z}{\partial c_s} = \sum_{t=s}^{T} \frac{\partial z_t}{\partial c_s}$.

A critical iterative property is adopted to calculate the above value:

$$\frac{\partial z}{\partial c_{s-1}} = \frac{\partial z}{\partial c_s}\frac{\partial c_s}{\partial c_{s-1}} + \frac{\partial z_{s-1}}{\partial c_{s-1}}. \tag{35}$$

A few other gradients can be calculated through the chain rule using the above calculation output:

$$\frac{\partial z_t}{\partial i_t} = \frac{\partial z_t}{\partial c_t}\frac{\partial c_t}{\partial i_t}, \frac{\partial c_t}{\partial i_t} = g_t \tag{36}$$

$$\frac{\partial z_t}{\partial f_t} = \frac{\partial z_t}{\partial c_t}\frac{\partial c_t}{\partial f_t}, \frac{\partial c_t}{\partial f_t} = c_{t-1} \tag{37}$$

$$\frac{\partial z_t}{\partial o_t} = \frac{\partial z_t}{\partial h_t}\frac{\partial h_t}{\partial o_t}, \frac{\partial h_t}{\partial o_t} = tanh(c_t) \tag{38}$$

$$\frac{\partial z_t}{\partial g_t} = \frac{\partial z_t}{\partial c_t}\frac{\partial c_t}{\partial g_t}, \frac{\partial c_t}{\partial g_t} = i_t \tag{39}$$