E-prop maths

Werner van der Veen

November 5, 2020

1 Proof: BPTT to E-prop

The main equation to be proved:

$$\frac{dE}{dW_{ji}} = \sum_t \frac{dE}{dz_j^t} \cdot \left[\frac{dz_j^t}{dW_{ji}}\right]_{\text{local}} \tag{1}$$

We start with the classical factorization of the loss gradients in an unrolled RNN:

$$\frac{dE}{dW_{ji}} = \frac{dE}{d\mathbf{h}_j^{t'}} \cdot \frac{\partial \mathbf{h}_j^{t'}}{\partial W_{ji}} \tag{2}$$

The summation indicates that weights are shared in an unrolled RNN.

We now decompose the first term into a series of learning signals $L_j^t = \frac{dE}{dz_j^t}$ and local factors $\frac{\partial \mathbf{h}_j^{t-t'}}{\partial \mathbf{h}_j^t}$ for t since the event horizon t':

$$\frac{dE}{d\mathbf{h}_j^{t'}} = \underbrace{\frac{dE}{dz_j^{t'}}}_{L_j^{t'}} \frac{\partial z_j^{t'}}{\partial \mathbf{h}_j^{t'}} + \frac{dE}{d\mathbf{h}_j^{t'+1}} \frac{\partial \mathbf{h}_j^{t'+1}}{\partial \mathbf{h}_j^{t'}} \tag{3}$$

Note that this equation is recursive. If we substitute the equation (3) into the classical factorization (2), we get:

$$\frac{dE}{dW_{ji}} = \sum_{t'} \left(L_j^{t'} \frac{\partial z_j^{t'}}{\partial \mathbf{h}_j^{t'}} + \frac{dE}{d\mathbf{h}_j^{t'+1}} \frac{\partial \mathbf{h}_j^{'t+1}}{\partial \mathbf{h}_j^{'t}} \right) \cdot \frac{\partial \mathbf{h}_j^{t'}}{\partial W_{ji}} \tag{4}$$

$$= \sum_{t'} \left(L_j^{t'} \frac{\partial z_j^{t'}}{\partial \mathbf{h}_j^{t'}} + \left(L_j^{t'+1} \frac{\partial z_j^{t'+1}}{\partial \mathbf{h}_j^{t'+1}} + (\cdots) \frac{\partial \mathbf{h}_j^{t'+2}}{\partial \mathbf{h}_j^{t'+1}} \right) \frac{\partial \mathbf{h}_j^{'t+1}}{\partial \mathbf{h}_j^{'t}} \right) \cdot \frac{\partial \mathbf{h}_j^{t'}}{\partial W_{ji}} \tag{5}$$

We write the term in parentheses into a second term indexed by t:

$$\frac{dE}{dW_{ji}} = \sum_{t'} \sum_{t \geq t'} L_j^t \frac{\partial z_j^t}{\partial \mathbf{h}_j^t} \frac{\partial \mathbf{h}_j^t}{\partial \mathbf{h}_j^{t-1}} \cdots \frac{\partial \mathbf{h}_j^{t+1}}{\partial \mathbf{h}_j^{t'}} \cdot \frac{\partial \mathbf{h}_j^{'t}}{\partial W_{ji}} \tag{6}$$

We then exchange the summation indices to pull out the learning signal L_j^t. This expresses the loss as a sum of learning signals multiplied by something we define as the eligibility trace. This eligibility trace consists of $\frac{\partial z_j^t}{\partial \mathbf{h}_j^t}$ and the eligibility vector ϵ_{ji}^t:

$$\frac{dE}{dW_{ji}} = \sum_t L_j^t \underbrace{\frac{\partial z_j^t}{\partial \mathbf{h}_j^t} \underbrace{\sum_{t \geq t'} \frac{\partial \mathbf{h}_j^t}{\partial \mathbf{h}_j^{t-1}} \cdots \frac{\partial \mathbf{h}_j^{t+1}}{\partial \mathbf{h}_j^{t'}} \cdot \frac{\partial \mathbf{h}_j^{'t}}{\partial W_{ji}}}_{\epsilon_{ji}^t}}_{e_{ji}^t} \tag{7}$$

This is the main e-prop equation.

2 Single-layer e-prop in pseudocode (LIF)

In LIF, $\{\mathbf{h}_j^t, \epsilon_{ji}^t\} \subset \mathbb{R}$.

```
for $t$ in $T$ do
    $z_j^t \leftarrow \begin{cases} 0, & \text{if } t - t_{z_j} < \delta t_{\text{ref}}. \\ H(v_j^t - v_{\text{th}}), & \text{otherwise.} \end{cases}$
    $I_j^t \leftarrow \sum_i W_{ji} z_i^t + \sum_u W_{ju} u(t)$
    $v_j^{t+1} \leftarrow \alpha v_j^t + I_j^t - z_j^t \alpha v_j^t - z_j^{t-\delta t_{\text{ref}}} \alpha v_j^t$
    $\epsilon_{ji}^{t+1} = \alpha(1 - z_j - z_j^{t-\delta t_{\text{ref}}}) \epsilon_{ji}^t + z_i^t$
    $h_j^{t+1} \leftarrow \begin{cases} -\gamma, & \text{if } t - t_{z_j} < \delta t_{\text{ref}}. \\ \gamma \max\left(0, 1 - \left|\frac{v_j^{t+1} - v_{\text{th}}}{v_{\text{th}}}\right|\right), & \text{otherwise.} \end{cases}$
    $e_{ji}^{t+1} \leftarrow h_j^{t+1} \epsilon_{ji}^{t+1}$
    $y_k^t = \kappa y_k^{t-1} + \sum_j W_{kj}^{\text{out}} z_j^t + b_k^{\text{out}}$
    $W \leftarrow W - \eta \sum_t \left(\sum_k B_{jk} \left(y_k^t - y_k^{*,t}\right)\right) e_{ji}^t$
end for
```

3 ALIF steps

$$I_j^t = \sum_{t \neq j} W_{ji}^{\text{rec}} z_i^t + \sum_i W_{ji}^{\text{in}} x_i^{t+1} \tag{8}$$

$$A_j^t = v_{\text{th}} + \beta a_j^t \tag{9}$$

$$z_j^t = H\left(v_j^t - A_j^t\right) \tag{10}$$

$$\psi_j^t = \frac{1}{v_{\text{th}}} 0.3 \max\left(0, 1 - \left|\frac{v_j^t - A_j^t}{v_{\text{th}}}\right|\right) \tag{11}$$

$$y_k^t = \kappa y_k^{t-1} + \sum_j W_{kj}^{\text{out}} z_j^t + b_k^{\text{out}} \tag{12}$$

$$e_{ji}^t = \psi_j^t \left(\epsilon_{ji,v}^t - \beta \epsilon_{ji,a}^t\right) \tag{13}$$

$$\bar{e}_{ji}^t = \kappa \bar{e}_{ji}^{t-1} + e_{ji}^t \tag{14}$$

$$v_j^{t+1} = \alpha v_j^t + I_j^t - z_j v_{\text{th}} \tag{15}$$

$$a_j^{t+1} = \rho a_j^t + z_j^t \tag{16}$$

$$\epsilon_{ji,v}^{t+1} = \alpha \epsilon_{ji,v}^t + z_i^t \tag{17}$$

$$\epsilon_{ji,a}^{t+1} = \psi_j^t \epsilon_{ji,v}^t + \left(\rho - \psi_j^t \beta\right) \epsilon_{ji,a}^t \tag{18}$$

$$\Delta W_{ji}^{\text{rec}} = -\eta \sum_t \left(\sum_k B_{jk} \left(y_k^t - y_k^{*,t}\right)\right) \bar{e}_{ji}^t \tag{19}$$

$$\tag{20}$$

Given at time t, with given observable state z_i^t (simplified):

$$v_j^{t+1} = \alpha v_j^t + \sum_{t \neq j} W_{ji}^{\text{rec},t} z_i^t + \sum_i W_{ji}^{\text{in}} x_i^{t+1} - H\left(v_j^t - v_{\text{th}} - \beta a_j^t\right) v_{\text{th}} \tag{21}$$

$$a_j^{t+1} = \rho a_j^t + H\left(v_j^t - v_{\text{th}} - \beta a_j^t\right) \tag{22}$$

$$\epsilon_{ji,v}^{t+1} = \alpha \epsilon_{ji,v}^t + z_i^t \tag{23}$$

$$\epsilon_{ji,a}^{t+1} = \frac{1}{v_{\text{th}}} 0.3 \max\left(0, 1 - \left|\frac{v_j^t - v_{\text{th}} - \beta a_j^t}{v_{\text{th}}}\right|\right) \epsilon_{ji,v}^t \tag{24}$$

$$+ \left(\rho - \frac{1}{v_{\text{th}}} 0.3 \max\left(0, 1 - \left|\frac{v_j^t - v_{\text{th}} - \beta a_j^t}{v_{\text{th}}}\right|\right) \beta\right) \epsilon_{ji,a}^t \tag{25}$$

$$\bar{e}_{ji}^t = \kappa \bar{e}_{ji}^{t-1} + \frac{1}{v_{\text{th}}} 0.3 \max\left(0, 1 - \left|\frac{v_j^t - v_{\text{th}} - \beta a_j^t}{v_{\text{th}}}\right|\right) \left(\epsilon_{ji,v}^t - \beta \epsilon_{ji,a}^t\right) \tag{26}$$

$$y_k^t = \kappa y_k^{t-1} + \sum_j W_{kj}^{\text{out}} z_j^t + b_k^{\text{out}} \tag{27}$$

$$W_{ji}^{\text{rec},t+1} = W_{ji}^{\text{rec},t} - \eta \sum_t \left(\sum_k B_{jk} \left(y_k^t - y_k^{*,t}\right)\right) \bar{e}_{ji}^t \tag{28}$$

$$\tag{29}$$

Effects of $\mathbf{h}$ on W:

$$\frac{\partial v_j^t}{\partial W_{ji}} = \epsilon_{ji,v}^{t-1} \tag{30}$$

$$\frac{\partial u_j^t}{\partial W_{ji}} = 0 \tag{31}$$

4 Audio preprocessing

Goal: turn a .wav signal consisting of N phones into N chunks.

Frame length = 400 samples.

Frame step = 160 samples.

For every frame, 13 MFCC coefficients are extracted.

We take the DFT:

$$S_i(k) = \sum_{n=1}^{N} s_i(n)h(n)e^{-j2\pi kn/N} \qquad 1 \leq k \leq K \tag{32}$$

where $h(n)$ is an N sample long analysis window (e.g. hamming window), and K is the length of the DFT (in our case, $K = 512$).

We then take the periodogram estimate.

$$P_i(k) = \frac{1}{N} \left|S_i(k)\right|^2 \tag{33}$$

where we then keep only the first 257 coefficients.

We have a filterbank of 26 triangular filters of length 257. Each filter is mostly zero, except its specific region.

To compute the Mel filterbank, we choose lower and upper frequencies of 0Hz and 8000Hz and we convert them to Mels using

$$M(f) = 1125 \ln(1 + f/700) \tag{34}$$

We obtain 0 Mels and 2834.99 Mels.

To obtain our 26 filterbanks, we need 26 points spaced linearly between 0 and 2834.99:

```
[   0.        ,  104.99962963,  209.99925926,  314.99888889,
  419.99851852,  524.99814815,  629.99777778,  734.99740741,
  839.99703704,  944.99666667, 1049.9962963 , 1154.99592593,
 1259.99555556, 1364.99518519, 1469.99481481, 1574.99444444,
 1679.99407407, 1784.9937037 , 1889.99333333, 1994.99296296,
 2099.99259259, 2204.99222222, 2309.99185185, 2414.99148148,
 2519.99111111, 2624.99074074, 2729.99037037, 2834.99      ]
```

We then convert these back to Hertz using

$$M^{-1}(m) = 700\left(\exp(m/1125) - 1\right) \tag{35}$$

and obtain

```
[   0.        ,   68.47907878,  143.65727789,  226.18995388,
  316.79657505,  416.26699328,  525.46832953,  645.35253278,
  776.96467861,  921.45207944, 1080.07428613, 1254.21406795,
 1445.38946668, 1655.26702996, 1885.67633923, 2138.62595885,
 2416.32094551, 2721.18207055, 3055.86692273, 3423.29307542,
 3826.6635202 , 4269.49458849, 4755.6466048 , 5289.35753858,
 5875.27994818, 6518.52153898, 7224.68968918, 7999.94033136]
```

We round these frequencies to the nearest FFT bin, using

$$f(h) = \lfloor (512+1)h/16000 \rfloor \tag{36}$$

resulting in the sequence

```
[  0.,   2.,   4.,   7.,  10.,  13.,  16.,  20.,  24.,  29.,  34.,
  40.,  46.,  53.,  60.,  68.,  77.,  87.,  97., 109., 122., 136.,
 152., 169., 188., 209., 231., 256.]
```

We can use this to finally create our Mel-spaced filterbank:

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \frac{k-f(m-1)}{f(m)-f(m-1)} & f(m-1) \le k \le f(m) \\ \frac{f(m+1)-k}{f(m+1)-f(m)} & f(m) \le k \le f(m+1) \\ 0 & k > f(m+1) \end{cases} \tag{37}$$

To calculate filterbank energies, we multiply each filterbank with the power spectrum, then add up the coefficients. We then obtain 26 values that indicate how much energy was in each filterbank.

We then take the log of each of these 26 energies.

Finally, we take the DCT of the 26 log filterbank energies to obtain 26 cepstral coefficients. We then retain only the first 13 of the coefficients for each frame.

We also compute the first and second derivatives (delta and delta-delta coefficients, resp.) of the 13 coefficients using

$$d_t = \frac{\sum_{n=1}^{N} n\,(c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2} \tag{38}$$

where d_t is a delta coefficient from frame t in terms of the static coefficients c_{t+n} to c_{t-n}, using $N = 2$:

$$d_t = \frac{c_{t+1} - c_{t-1} + 2c_{t+2} - 2c_{t-2}}{10} \tag{39}$$

Second derivatives are calculated from the deltas, not the static coefficients.

We finally concatenate the zeroth, first, and second derivatives and acquire a vector of length 39.

5 Bellec's TIMIT implementatation

Key points:

- $N \times 39$ audio frames and
  $N \times 61$ class labels.
- Bi-directional LSNN.
- 300 LIF, 100 ALIF, $\theta_m = 20$ ms, $\theta_a = 200$ ms, $\beta = 0.184$, $v_{\text{th}} = 1.6$, $\theta_{\text{out}} = 3$ ms, refractory period of 2 ms.
- Train, validation, test size = 3696, 400, 192 sequences. Input $\mathbf{x}^t$ is [0,1]–normalized concatenation of zeroth to second derivatives of MFCCs.
- Each frame in the sequence is fed for 5 consecutive 1 ms steps.
- Networks trained for 80 epochs with early stopping on lowest validation error.
- Adam optimizer (with defaults except $\epsilon_{\text{Adam}} = 10^{-5}$.
- Batch size 32, $\gamma = 10^{-2}$. Adaptive B_{jk} initialized as Gaussian with $\mu = 0$ and $\sigma = 1/n$ with n being neurons in LSNN.
- L2 regularization with additional term $10^{-5} \cdot \|W\|^2$.
- In adaptive B, L2 weight decay on output and broadcast neurons according to [2] using $c_{\text{decay}} = 10^{-2}$.
- Fire rate regularization with $c_{\text{reg}} = 50$.

Procedure:

```
For 80 epochs:

For 32 batches:

F = get_frame_from_

For 5 steps:

L = M(F)
```