Deep Learning Class
Homework 1

Anton Babenko

June 13, 2021

1. Network description

This homework contains full description of forward(scalar and vector forms) and backward passes(scalar form) of AlarmworkNet, which has three layers: $Layer_{in1}$, $Layer_{in2}$ and $Layer_{out}$. This network is a modified version of Simple Recurrent Network, which is also known as Elman Network.
The loss function for this network is Sum Squared Error:

$$w^* \leftarrow \underset{w}{\operatorname{argmin}} \frac{1}{2} \sum_{n=1}^{N} (t_{\{n\}} - y_{\{n\}}(\hat{x}_{\{n\}}, w))^2,$$

where $t_{\{n\}}$ is target value from the training dataset, $1 \leq n \leq N$, N is number of samples in the training dataset; $\hat{x} = \{x_1, x_2, ...x_L\}$ is a sequence of L input vectors, where each input vector x_l, $1 \leq l \leq L$ has dimension D. Neural network is trained to produce output vectors y of dimension M.
Also, $Layer_{in1}$ and $Layer_{in2}$ contains P neurons each.

2. Problem 1 - Forward pass

This part contains the description of the forward pass of the neural network.

(1) Let's write down a dimensionalities of values, weights and biases:

(a) Values:

1) x, where $dim(x) = D$
2) y, where $dim(y) = M$
3) $a^{(in1)}$, where $dim(a^{(in1)}) = P$
4) $z^{(in1)}$, where $dim(z^{(in1)}) = P$
5) $a^{(in2)}$, where $dim(a^{(in2)}) = P$

6) $z^{(in2)}$, where $dim(z^{(in2)}) = P$
7) $z^{(in12)}$, where $dim(z^{(in12)}) = P$
8) $a^{(out)}$, where $dim(a^{(out)}) = M$
9) $z^{(out)}$, where $dim(z^{(out)}) = M$

(b) Weights and biases:

1) $w^{(in1)}$, where $dim(w^{(in1)}) = P \times D$
2) $w^{(rec1)}$, where $dim(w^{(rec1)}) = P \times P$
3) $b^{(in1)}$, where $dim(b^{(in1)}) = P$
4) $w^{(in2)}$, where $dim(w^{(in2)}) = P \times D$
5) $w^{(rec2)}$, where $dim(w^{(rec2)}) = P \times P$
6) $b^{(in2)}$, where $dim(b^{(in2)}) = P$
7) $w^{(out)}$, where $dim(w^{(out)}) = M \times P$
8) $b^{(out)}$, where $dim(b^{(out)}) = M$

(2) Let's write down forward equations for each layer in a scalar form:

(a) $Layer_{in1}$

$$a_j^{(in1)}[l] = \sum_{i=1}^{D} (w_{ji}^{(in1)} x_i) + \sum_{i=1}^{P} (w_{ji}^{(rec1)} z_i^{(in12)}[l]) + b_j^{(in1)}$$
$$z_j^{(in1)}[l] = \tanh(a_j^{(in1)}[l])$$

, where $1 \leq j \leq P$

(b) $Layer_{in2}$

$$\begin{cases} \left[\begin{array}{l} a_j^{(in2)}[l] = \sum_{i=1}^{D} (w_{ji}^{(in2)} x_i) + \sum_{i=1}^{P} (w_{ji}^{(rec2)} z_i^{(in2)}[l-1]) + b_j^{(in2)} \\ z_j^{(in2)}[l] = \tanh(a_j^{(in2)}[l]) \end{array} \right. & \text{if } l\%2 == 0 \\ z_j^{(in2)}[l] = z_j^{(in2)}[l-1] & \text{if } l\%2 == 1 \end{cases}$$

, where $1 \leq j \leq P$

(c) $Layer_{out}$

$$a_j^{(out)}[l] = \sum_{i=1}^{P} (w_{ji}^{(out)} z_i^{(in1)}[l]) + b_j^{(out}$$
$$z_j^{(out)}[l] = \tanh(a_j^{(out)}[l])$$

, where $1 \leq j \leq M$

(3) Let's write down forward equations for each layer in a vector form:

(a) $Layer_{in1}$

$$a^{(in1)}[l] = w^{(in1)}x + w^{(rec1)}z^{(in12)}[l] + b^{(in1)}$$
$$z^{(in1)}[l] = \tanh(a^{(in1)}[l])$$

(b) $Layer_{in2}$

$$\begin{cases} \left[\begin{array}{l} a^{(in2)}[l] = w^{(in2)}x + w^{(rec2)}z^{(in2)}[l-1] + b^{(in2)} \\ z^{(in2)} = \tanh(a^{(in2)}[l]) \end{array} \right. & \text{if } l\%2 == 0 \\ z^{(in2)}[l] = z^{(in2)}[l-1] & \text{if } l\%2 == 1 \end{cases}$$

(c) $Layer_{out}$

$$a^{(out)}[l] = w^{(out)}z^{(in1)}[l] + b^{(out}$$
$$z^{(out)}[l] = \tanh(a^{(out)}[l])$$

3. Problem 2 - Backward pass

This part contains the description of the backward pass of the neural network.

(1) Let's write down the derivatives for neural network's output layer weights updates:

$$\frac{\partial Loss}{\partial w^{(out)}} = z^{(in1)}[l](t - \tanh(w^{(out)}z^{(in1)}[l] + b^{(out)}))(\tanh^2(w^{(out)}z^{(in1)}[l] + b^{(out)}) - 1)$$
$$\frac{\partial Loss}{\partial b^{(out)}} = (t - \tanh(w^{(out)}z^{(in1)}[l] + b^{(out)}))(\tanh^2(w^{(out)}z^{(in1)}[l] + b^{(out)}) - 1)$$

(2) Let's write down the backward pass flow for local gradients δ for each layer in scalar form:

(a) $Layer_{out}$

$$\frac{\partial Loss}{\partial w_{ji}^{(out)}} = (t_j - y_j)(\tanh^2(a_j^{(out)}[l]) - 1)z_i^{(in1)} \tag{3.1}$$
$$\frac{\partial Loss}{\partial b_j^{(out)}} = (t_j - y_j)(\tanh^2(a_j^{(out)}[l]) - 1) \tag{3.2}$$

, where $1 \leq j \leq M$, $1 \leq i \leq P$.
From the previous equations(3.1, 3.2) we can define delta part to simplify derivatives in the next formulas:

$$\delta_j^{(out)}[l] = (t_j - y_j)(\tanh^2(a_j^{(out)}[l]) - 1) \tag{3.3}$$

, where $1 \leq j \leq M$.
Hence, based on $\delta^{(out)}$(3.3), we have:

$$\frac{\partial Loss}{\partial w_{ji}^{(out)}} = \delta_j^{(out)}[l] z_i^{(in1)}[l]$$
$$\frac{\partial Loss}{\partial b_j^{(out)}} = \delta_j^{(out)}[l]$$

, where $1 \leq j \leq M$, $1 \leq i \leq P$.

(b) $Layer_{in1}$
To simplify formulas, let's use $\delta^{(out)}$(3.3) in the next equations:

$$\frac{\partial Loss}{\partial w_{ik}^{(in1)}} = (1 - \tanh^2(a_i^{(in1)}[l])) \sum_{j=1}^{M} \delta_{ji}^{(out)}[l] w_{ji}^{(out)} x_k \tag{3.4}$$
$$\frac{\partial Loss}{\partial b_i^{(in1)}} = (1 - \tanh^2(a_i^{(in1)}[l])) \sum_{j=1}^{M} \delta_j^{(out)}[l] w_{ji}^{(out)} \tag{3.5}$$
$$\frac{\partial Loss}{\partial w_{ik}^{(rec1)}} = (1 - \tanh^2(a_i^{(in1)}[l])) \sum_{j=1}^{M} \delta_{ji}^{(out)}[l] w_{ji}^{(out)} z_k^{(in12)}[l] \tag{3.6}$$

, where $1 \leq i \leq P$, $1 \leq k \leq D$.
From the previous equations(3.4, 3.5, 3.6) we can define delta part to simplify derivatives in the next formulas:

$$\delta_i^{(in1)}[l] = (1 - \tanh^2(a_i^{(in1)}[l])) \sum_{j=1}^{M} \delta_{ji}^{(out)}[l] w_{ji}^{(out)} \tag{3.7}$$

, where $1 \leq i \leq P$.
Hence, based on $\delta^{(in1)}$(3.7), we have:

$$\frac{\partial Loss}{\partial w_{ik}^{(in1)}} = \delta_i^{(in1)}[l] x_k$$
$$\frac{\partial Loss}{\partial b_i^{(in1)}} = \delta_i^{(in1)}[l]$$
$$\frac{\partial Loss}{\partial w_{ik}^{(rec1)}} = \delta_i^{(in1)}[l] z_k^{(in12)}[l]$$

, where $1 \leq i \leq P$, $1 \leq k \leq D$.

(c) $Layer_{in2}$
To simplify formulas, let's use $\delta^{(in1)}$(3.7) in the next equations:

$$\frac{\partial Loss}{\partial w_{nd}^{(in2)}} = \sum_{i=1}^{P} (\delta_i^{(in1)}[l] w_{in}^{(rec1)})(1 - \tanh^2(a_n^{(in2)}[l])) x_d \tag{3.8}$$

$$\frac{\partial Loss}{\partial b_n^{(in2)}} = \sum_{i=1}^{P} (\delta_i^{(in1)}[l] w_{in}^{(rec1)})(1 - \tanh^2(a_n^{(in2)}[l])) \tag{3.9}$$

$$\frac{\partial Loss}{\partial w_{nm}^{(rec2)}} = \sum_{i=1}^{P} (\delta_i^{(in1)}[l] w_{in}^{(rec1)})(1 - \tanh^2(a_n^{(in2)}[l])) z_m^{(in2)}[l-1] \tag{3.10}$$

, where $1 \leq n, m \leq P$, $1 \leq d \leq D$.
From the previous equations(3.8, 3.9, 3.10) we can define delta part to simplify derivatives in the next formulas:

$$\delta_n^{(in2)}[l] = \sum_{i=1}^{P} (\delta_i^{(in1)}[l] w_{in}^{(rec1)})(1 - \tanh^2(a_n^{(in2)}[l])) \tag{3.11}$$

, where $1 \leq n \leq P$
Hence, based on $\delta^{(in2)}$(3.11), we have:

$$\frac{\partial Loss}{\partial w_{nd}^{(in2)}} = \delta_n^{(in2)}[l] x_d$$

$$\frac{\partial Loss}{\partial b_n^{(in2)}} = \delta_n^{(in2)}[l]$$

$$\frac{\partial Loss}{\partial w_{nm}^{(rec2)}} = \delta_n^{(in2)}[l] z_m^{(in2)}[l-1]$$

, where $1 \leq n, m \leq P$, $1 \leq d \leq D$.