# 1 Preprocess

Alphabet Size: $|\mathcal{A}| = 4$
Character to One-hot Encoding:
Given a vocabulary $\mathcal{A} = \{h, e, l, o\}$, the one-hot encoding of a character $c \in \mathcal{A}$ is defined as:

$$\text{one\_hot}(c)_i = \begin{cases} 1 & \text{if } \mathcal{A}_i = c, \\ 0 & \text{otherwise.} \end{cases}$$

$$\begin{aligned} h &: [1\ 0\ 0\ 0] \\ e &: [0\ 1\ 0\ 0] \\ l &: [0\ 0\ 1\ 0] \\ o &: [0\ 0\ 0\ 1] \end{aligned}$$

Input and Target Sequences:

$$\begin{aligned} x_1 &= h & y_1 &= e \\ x_2 &= e & y_2 &= l \\ x_3 &= l & y_3 &= l \\ x_4 &= l & y_4 &= o \end{aligned}$$

Replacing characters with one-hot encoding:

$$\begin{aligned} x_1^o &= [1\ 0\ 0\ 0] & y_1 &= [0\ 1\ 0\ 0] \\ x_2^o &= [0\ 1\ 0\ 0] & y_2 &= [0\ 0\ 1\ 0] \\ x_3^o &= [0\ 0\ 1\ 0] & y_3 &= [0\ 0\ 1\ 0] \\ x_4^o &= [0\ 0\ 1\ 0] & y_4 &= [0\ 0\ 0\ 1] \end{aligned}$$

# 2 Forward Pass Formulas

For a sequence of characters $x_1, x_2, \ldots, x_T$, the network computes:

1. Embeding Layer for computing $x_1, x_2, \ldots, x_T$

$$x_t = x_t^o E$$

2. Hidden State at time $t$, $h_t$:

$$h_t = \tanh(Ux_t + Wh_{t-1} + b)$$

3. Output before softmax, $\Omega_t$:

$$\Omega_t = Vh_t + c$$

4. Softmax Output, $\hat{y}_t$, for each character:

$$\text{softmax}(\Omega_t) = \hat{y}_t = \frac{e^{\Omega_t}}{\sum_{k=1}^{|\mathcal{A}|} e^{\Omega_{t,k}}} \text{ for } t = 1, \ldots, |\mathcal{A}|$$

5. Cross-Entropy Loss for the correct character $y_t$:

$$L_t = -\ y_t\ ln(\hat{y}_t)$$

# 3 Backpropagation Through Time Formulas

Gradients of the loss $L$ with respect to the parameters $U, W, V, b, c$ are computed as follows:

1. Gradient of Loss w.r.t. Output (Softmax Gradient):

$$\frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial \Omega_t} = (\hat{y}_t - y_t)$$

2. Updates for $V$ and $c$:

$$\begin{aligned}
\frac{\partial L}{\partial V} &= \sum_{t=1}^{S} \frac{\partial L_t}{\partial V} \\
&= \sum_{t=1}^{S} \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial \Omega_t} \frac{\partial \Omega_t}{\partial V} \\
&= \sum_{t=1}^{S} (\hat{y}_t - y_t) \cdot h_t^{\dagger} \\
\frac{\partial L}{\partial c} &= \sum_{t=1}^{T} \frac{\partial L_t}{\partial c} \\
&= \sum_{t=1}^{T} \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial \Omega_t} \frac{\partial \Omega_t}{\partial c} \\
&= \sum_{t=1}^{T} (\hat{y}_t - y_t)
\end{aligned}$$

3. Updates for $U$, $W$, $b$ and $E$:

$$\begin{aligned}
\frac{\partial L}{\partial W} &= \sum_{t=1}^{\mathcal{S}} \left( \sum_{k=1}^{t} \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial \Omega_t} \frac{\partial \Omega_t}{\partial h_t} \prod_{j=k}^{t-1} \left( \frac{\partial h_{j+1}}{\partial h_j} \right) \frac{\partial h_k}{\partial W} \right) \\
&= \sum_{t=1}^{\mathcal{S}} \left( \sum_{k=1}^{t} (y_t - \hat{y}_t)\ V^\dagger \prod_{j=k}^{t-1} \left( W^\dagger\ (1 - h_{j+1}^2) \right) (1 - h_k^2)\ h_{k-1} \right) \\
\frac{\partial L}{\partial U} &= \sum_{t=1}^{\mathcal{S}} \left( \sum_{k=1}^{t} \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial \Omega_t} \frac{\partial \Omega_t}{\partial h_t} \prod_{j=k}^{t-1} \left( \frac{\partial h_{j+1}}{\partial h_j} \right) \frac{\partial h_k}{\partial U} \right) \\
&= \sum_{t=1}^{\mathcal{S}} \left( \sum_{k=1}^{t} (y_t - \hat{y}_t)\ V^\dagger \prod_{j=k}^{t-1} \left( W^\dagger\ (1 - h_{j+1}^2) \right) (1 - h_k^2)\ x_k \right) \\
\frac{\partial L}{\partial b} &= \sum_{t=1}^{\mathcal{S}} \left( \sum_{k=1}^{t} \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial \Omega_t} \frac{\partial \Omega_t}{\partial h_t} \prod_{j=k}^{t-1} \left( \frac{\partial h_{j+1}}{\partial h_j} \right) \frac{\partial h_k}{\partial b} \right) \\
&= \sum_{t=1}^{\mathcal{S}} \left( \sum_{k=1}^{t} (y_t - \hat{y}_t)\ V^\dagger \prod_{j=k}^{t-1} \left( W^\dagger\ (1 - h_{j+1}^2) \right) (1 - h_k^2) \right) \\
\frac{\partial L}{\partial E} &= \sum_{t=1}^{\mathcal{S}} \left( \sum_{k=1}^{t} \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \prod_{j=k}^{t-1} \left( \frac{\partial h_{j+1}}{\partial h_j} \right) \frac{\partial h_k}{\partial x_k} \frac{\partial x_k}{\partial E} \right) \\
&\quad \sum_{t=1}^{\mathcal{S}} \left( \sum_{k=1}^{t} (y_t - \hat{y}_t)\ V^\dagger \prod_{j=k}^{t-1} \left( W^\dagger\ (1 - h_{j+1}^2) \right) (1 - h_k^2)\ U\ x_k^o \right)
\end{aligned}$$

## Parameter Updates

The parameters are updated by subtracting the gradient scaled by a learning rate $\eta$:

$$\begin{aligned}
V &= V - \eta \frac{\partial L}{\partial V} \\
c &= c - \eta \frac{\partial L}{\partial c} \\
W &= W - \eta \frac{\partial L}{\partial W} \\
U &= U - \eta \frac{\partial L}{\partial U} \\
b &= b - \eta \frac{\partial L}{\partial b} \\
E &= E - \eta \frac{\partial L}{\partial E}
\end{aligned}$$

# 4 Parameters Initialized

The network parameters are initialized as follows:

$$E = \begin{bmatrix} 0.8175 & 0.4613 \\ -0.3599 & -0.4824 \\ 0.1268 & 0.1540 \\ -0.0358 & -0.5677 \end{bmatrix}$$

$$U = \begin{bmatrix} 0.1442 & -0.2315 \end{bmatrix}$$

$$W = \begin{bmatrix} -0.5352 \end{bmatrix}$$

$$b = \begin{bmatrix} 0. \end{bmatrix}$$

$$V = \begin{bmatrix} 0.6951 \\ -0.4402 \\ -0.2246 \\ -0.3053 \end{bmatrix}$$

$$c = \begin{bmatrix} 0. \\ 0. \\ 0. \\ 0. \end{bmatrix}$$

# 5 Forward Pass

## 5.1 Step 1

$$
\begin{aligned}
x_1 &= x_1^o \cdot E \\
&= [1\ 0\ 0\ 0] \cdot \begin{bmatrix} 0.8175 & 0.4613 \\ -0.3599 & -0.4824 \\ 0.1268 & 0.1540 \\ -0.0358 & -0.5677 \end{bmatrix} \\
x_1 &= \begin{bmatrix} 0.8175 & 0.4613 \end{bmatrix} \\
h_1 &= \tanh(U \cdot x_1 + W \cdot h_0 + b) \\
&= \tanh\left( \begin{bmatrix} 0.1442 & -0.2315 \end{bmatrix} \cdot \begin{bmatrix} 0.8175 \\ 0.4613 \end{bmatrix} + [-0.5352] \cdot [0] + [0] \right) \\
h_1 &= \begin{bmatrix} 0.0111 \end{bmatrix} \\
\Omega_1 &= V \cdot h_1 + c \\
&= \begin{bmatrix} 0.6951 \\ -0.4402 \\ -0.2246 \\ -0.3053 \end{bmatrix} \cdot \begin{bmatrix} 0.0111 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
\Omega_1 &= \begin{bmatrix} 0.0077 \\ -0.0049 \\ -0.0025 \\ -0.0034 \end{bmatrix} \\
\hat{y}_1 &= \text{softmax}(\Omega_1) \\
\hat{y}_1 &= \begin{bmatrix} 0.2521 \\ 0.2490 \\ 0.2496 \\ 0.2493 \end{bmatrix}
\end{aligned}
$$

Based on the maximum value of the softmax function, our model predicts it as 'h', but we actually want it to predict 'e'.

$$
\begin{aligned}
\mathrm{L}_1 &= -y_1 \cdot \ln(\hat{y}_1) \\
&= -\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \cdot \ln \begin{bmatrix} 0.2521 \\ 0.2490 \\ 0.2496 \\ 0.2493 \end{bmatrix} \\
&= 1.3904
\end{aligned}
$$

### 5.2 Step 2

$$
\begin{aligned}
x_2 &= x_2^o \cdot E \\
&= [0\ 1\ 0\ 0] \cdot \begin{bmatrix} 0.8175 & 0.4613 \\ -0.3599 & -0.4824 \\ 0.1268 & 0.1540 \\ -0.0358 & -0.5677 \end{bmatrix} \\
x_2 &= \begin{bmatrix} -0.3599 & -0.4824 \end{bmatrix} \\
h_2 &= \tanh(U \cdot x_2 + W \cdot h_1 + b) \\
&= \tanh\left( \begin{bmatrix} 0.1442 & -0.2315 \end{bmatrix} \cdot \begin{bmatrix} -0.3599 \\ -0.4824 \end{bmatrix} + \begin{bmatrix} -0.5352 \end{bmatrix} \cdot \begin{bmatrix} 0.0111 \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} \right) \\
h_2 &= \begin{bmatrix} 0.0537 \end{bmatrix} \\
\Omega_2 &= V \cdot h_2 + c \\
&= \begin{bmatrix} 0.6951 \\ -0.4402 \\ -0.2246 \\ -0.3053 \end{bmatrix} \cdot \begin{bmatrix} 0.0537 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
\Omega_2 &= \begin{bmatrix} 0.0373 \\ -0.0237 \\ -0.0120 \\ -0.0164 \end{bmatrix} \\
\hat{y}_2 &= \text{softmax}(\Omega_2) \\
\hat{y}_2 &= \begin{bmatrix} 0.2604 \\ 0.2450 \\ 0.2478 \\ 0.2468 \end{bmatrix}
\end{aligned}
$$

Based on the maximum value of the softmax function, our model predicts it as 'h', but we actually want it to predict 'l'.

$$
\begin{aligned}
\mathrm{L}_2 &= -y_2 \cdot \ln(\hat{y}_2) \\
&= -\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \cdot \ln \begin{bmatrix} 0.2604 \\ 0.2450 \\ 0.2478 \\ 0.2468 \end{bmatrix} \\
&= 1.3950
\end{aligned}
$$

### 5.3 Step 3

$$
\begin{aligned}
x_3 &= x_3^o \cdot E \\
&= [0\ 0\ 1\ 0] \cdot \begin{bmatrix} 0.8175 & 0.4613 \\ -0.3599 & -0.4824 \\ 0.1268 & 0.1540 \\ -0.0358 & -0.5677 \end{bmatrix} \\
x_3 &= \begin{bmatrix} 0.1268 & 0.1540 \end{bmatrix} \\
h_3 &= \tanh(U \cdot x_3 + W \cdot h_2 + b) \\
&= \tanh\left( \begin{bmatrix} 0.1442 & -0.2315 \end{bmatrix} \cdot \begin{bmatrix} 0.1268 \\ 0.1540 \end{bmatrix} + \begin{bmatrix} -0.5352 \end{bmatrix} \cdot \begin{bmatrix} 0.0537 \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} \right) \\
h_3 &= \begin{bmatrix} -0.0461 \end{bmatrix} \\
\Omega_3 &= V \cdot h_3 + c \\
&= \begin{bmatrix} 0.6951 \\ -0.4402 \\ -0.2246 \\ -0.3053 \end{bmatrix} \cdot \begin{bmatrix} -0.0461 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
\Omega_3 &= \begin{bmatrix} -0.0320 \\ 0.0203 \\ 0.0103 \\ 0.0140 \end{bmatrix} \\
\hat{y}_3 &= \text{softmax}(\Omega_3) \\
\hat{y}_3 &= \begin{bmatrix} 0.2413 \\ 0.2543 \\ 0.2517 \\ 0.2528 \end{bmatrix}
\end{aligned}
$$

Based on the maximum value of the softmax function, our model predicts it as 'e', but we actually want it to predict 'l'.

$$
\begin{aligned}
\mathrm{L}_3 &= -y_3 \cdot \ln(\hat{y}_3) \\
&= -\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \cdot \ln \begin{bmatrix} 0.2413 \\ 0.2543 \\ 0.2517 \\ 0.2528 \end{bmatrix} \\
&= 1.3793
\end{aligned}
$$

### 5.4 Step 4

$$
\begin{aligned}
x_4 &= x_4^o \cdot E \\
&= [0\ 0\ 1\ 0] \cdot \begin{bmatrix} 0.8175 & 0.4613 \\ -0.3599 & -0.4824 \\ 0.1268 & 0.1540 \\ -0.0358 & -0.5677 \end{bmatrix} \\
x_4 &= \begin{bmatrix} 0.1268 & 0.1540 \end{bmatrix} \\
h_4 &= \tanh(U \cdot x_4 + W \cdot h_3 + b) \\
&= \tanh\left( \begin{bmatrix} 0.1442 & -0.2315 \end{bmatrix} \cdot \begin{bmatrix} 0.1268 \\ 0.1540 \end{bmatrix} + \begin{bmatrix} -0.5352 \end{bmatrix} \cdot \begin{bmatrix} -0.0461 \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix} \right) \\
h_4 &= \begin{bmatrix} 0.0073 \end{bmatrix} \\
\Omega_4 &= V \cdot h_4 + c \\
&= \begin{bmatrix} 0.6951 \\ -0.4402 \\ -0.2246 \\ -0.3053 \end{bmatrix} \cdot \begin{bmatrix} 0.0073 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
\Omega_4 &= \begin{bmatrix} 0.0051 \\ -0.0032 \\ -0.0016 \\ -0.0022 \end{bmatrix} \\
\hat{y}_4 &= \text{softmax}(\Omega_4) \\
\hat{y}_4 &= \begin{bmatrix} 0.2514 \\ 0.2493 \\ 0.2497 \\ 0.2496 \end{bmatrix}
\end{aligned}
$$

Based on the maximum value of the softmax function, our model predicts it as 'h', but we actually want it to predict 'o'.

$$
\begin{aligned}
\mathrm{L}_4 &= -y_4 \cdot \ln(\hat{y}_4) \\
&= -\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \cdot \ln \begin{bmatrix} 0.2514 \\ 0.2493 \\ 0.2497 \\ 0.2496 \end{bmatrix} \\
&= 1.3880
\end{aligned}
$$

$$
\sum_{t=1}^{k} L_t = 1.3904 + 1.3950 + 1.3793 + 1.3880 = 5.5528
$$

# 6 Backpropagation Through Time

## 6.1 Gradient of L w.r.t. Output

$$
\begin{aligned}
\frac{\partial L_1}{\partial \hat{y}_1}\frac{\partial \hat{y}_1}{\partial \Omega_1} &= (\hat{y}_1 - y_1) \\
&= \begin{bmatrix} 0.2521 \\ 0.2490 \\ 0.2496 \\ 0.2493 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} 0.2521 \\ -0.7510 \\ 0.2496 \\ 0.2493 \end{bmatrix} \\
\frac{\partial L_2}{\partial \hat{y}_2}\frac{\partial \hat{y}_2}{\partial \Omega_2} &= (\hat{y}_2 - y_2) \\
&= \begin{bmatrix} 0.2604 \\ 0.2450 \\ 0.2478 \\ 0.2468 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} 0.2604 \\ 0.2450 \\ -0.7522 \\ 0.2468 \end{bmatrix} \\
\frac{\partial L_3}{\partial \hat{y}_3}\frac{\partial \hat{y}_3}{\partial \Omega_3} &= (\hat{y}_3 - y_3) \\
&= \begin{bmatrix} 0.2413 \\ 0.2543 \\ 0.2517 \\ 0.2528 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} 0.2413 \\ 0.2543 \\ -0.7483 \\ 0.2528 \end{bmatrix} \\
\frac{\partial L_4}{\partial \hat{y}_4}\frac{\partial \hat{y}_4}{\partial \Omega_4} &= (\hat{y}_4 - y_4) \\
&= \begin{bmatrix} 0.2514 \\ 0.2493 \\ 0.2497 \\ 0.2496 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} 0.2514 \\ 0.2493 \\ 0.2497 \\ -0.7504 \end{bmatrix}
\end{aligned}
$$

### 6.2 Update $V$

$$
\begin{aligned}
\frac{\partial L}{\partial V} &= \sum_{t=1}^{S} \frac{\partial L_t}{\partial V} \\
&= \sum_{t=1}^{S} (\hat{y}_t - y_t) \cdot h_t^{\dagger} \\
&= \begin{bmatrix} 0.0075 \\ -0.0051 \\ -0.0013 \\ -0.0011 \end{bmatrix} \\
\eta &= 0.1 \\
V_{new} &= V - \eta \frac{\partial L}{\partial V} \\
&= \begin{bmatrix} 0.6951 \\ -0.4402 \\ -0.2246 \\ -0.3053 \end{bmatrix} - 0.1 \cdot \begin{bmatrix} 0.0075 \\ -0.0051 \\ -0.0013 \\ -0.0011 \end{bmatrix} \\
V_{new} &= \begin{bmatrix} 0.6944 \\ -0.4397 \\ -0.2244 \\ -0.3051 \end{bmatrix}
\end{aligned}
$$

### 6.3 Update $c$:

$$\begin{aligned}
\frac{\partial L}{\partial c} &= \sum_{t=1}^{\mathcal{S}} \frac{\partial L_t}{\partial c} \\
&= \sum_{t=1}^{\mathcal{S}} (\hat{y}_t - y_t) \\
&= \begin{bmatrix} 1.0052 \\ -0.0025 \\ -1.0011 \\ -0.0016 \end{bmatrix} \\
\eta &= 0.1 \\
c_{new} &= c - \eta \frac{\partial L}{\partial c} \\
&= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} - 0.1 \cdot \begin{bmatrix} 1.0052 \\ -0.0025 \\ -1.0011 \\ -0.0016 \end{bmatrix} \\
c_{new} &= \begin{bmatrix} -0.1005 \\ 0.0002 \\ 0.1001 \\ 0.0002 \end{bmatrix}
\end{aligned}$$

### 6.4 Update $W$:

$$\begin{aligned}
\frac{\partial L}{\partial W} &= \sum_{t=1}^{\mathcal{S}} \left( \sum_{k=1}^{t} \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial \Omega_t} \frac{\partial \Omega_t}{\partial h_t} \prod_{j=k}^{t-1} \left( \frac{\partial h_{j+1}}{\partial h_j} \right) \frac{\partial h_k}{\partial W} \right) \\
&= \sum_{t=1}^{\mathcal{S}} \left( \sum_{k=1}^{t} (y_t - \hat{y}_t)\ V^{\dagger} \prod_{j=k}^{t-1} \left( W^{\dagger}\ (1 - h_{j+1}^2) \right) (1 - h_k^2)\ h_{k-1} \right) \\
&= \begin{bmatrix} -0.0082 \end{bmatrix} \\
\eta &= 0.1 \\
W_{new} &= W - \eta \frac{\partial L}{\partial W} \\
&= \begin{bmatrix} -0.5352 \end{bmatrix} - 0.1 \cdot \begin{bmatrix} -0.0082 \end{bmatrix} \\
W_{new} &= \begin{bmatrix} -0.5343 \end{bmatrix}
\end{aligned}$$

### 6.5 Update $U$:

$$\begin{aligned}
\frac{\partial L}{\partial U} &= \sum_{t=1}^{\mathcal{S}} \left( \sum_{k=1}^{t} \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial \Omega_t} \frac{\partial \Omega_t}{\partial h_t} \prod_{j=k}^{t-1} \left( \frac{\partial h_{j+1}}{\partial h_j} \right) \frac{\partial h_k}{\partial U} \right) \\
&= \sum_{t=1}^{\mathcal{S}} \left( \sum_{k=1}^{t} (y_t - \hat{y}_t)\ V^{\dagger} \prod_{j=k}^{t-1} \left( W^{\dagger}\ (1 - h_{j+1}^2) \right) (1 - h_k^2)\ x_k \right) \\
&= \begin{bmatrix} 0.2137 & 0.0982 \end{bmatrix} \\
\eta &= 0.1 \\
U_{new} &= U - \eta \frac{\partial L}{\partial U} \\
&= \begin{bmatrix} 0.1442 & -0.2315 \end{bmatrix} - 0.1 \cdot \begin{bmatrix} 0.2137 & 0.0982 \end{bmatrix} \\
U_{new} &= \begin{bmatrix} 0.1229 & -0.2413 \end{bmatrix}
\end{aligned}$$

### 6.6 Update $b$:

$$\begin{aligned}
\frac{\partial L}{\partial b} &= \sum_{t=1}^{\mathcal{S}} \left( \sum_{k=1}^{t} \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial \Omega_t} \frac{\partial \Omega_t}{\partial h_t} \prod_{j=k}^{t-1} \left( \frac{\partial h_{j+1}}{\partial h_j} \right) \frac{\partial h_k}{\partial b} \right) \\
&= \sum_{t=1}^{\mathcal{S}} \left( \sum_{k=1}^{t} (y_t - \hat{y}_t)\ V^{\dagger} \prod_{j=k}^{t-1} \left( W^{\dagger}\ (1 - h_{j+1}^2) \right) (1 - h_k^2) \right) \\
&= \begin{bmatrix} 0.7034 \end{bmatrix} \\
\eta &= 0.1 \\
b_{new} &= b - \eta \frac{\partial L}{\partial b} \\
&= \begin{bmatrix} 0 \end{bmatrix} - 0.1 \cdot \begin{bmatrix} 0.7034 \end{bmatrix} \\
b_{new} &= \begin{bmatrix} -0.0703 \end{bmatrix}
\end{aligned}$$

### 6.7 Update $E$:

$$\frac{\partial L}{\partial E} = \sum_{t=1}^{\mathcal{S}} \left( \sum_{k=1}^{t} \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial h_t} \prod_{j=k}^{t-1} \left( \frac{\partial h_{j+1}}{\partial h_j} \right) \frac{\partial h_k}{\partial x_k} \frac{\partial x_k}{\partial E} \right)$$

$$\sum_{t=1}^{\mathcal{S}} \left( \sum_{k=1}^{t} (y_t - \hat{y}_t)\, V^{\dagger} \prod_{j=k}^{t-1} \left( W^{\dagger}\, (1 - h_{j+1}^2) \right) (1 - h_k^2)\, U\, x_k^o \right)$$

$$= \begin{bmatrix} 0.0539 & -0.0865 \\ 0.0111 & -0.0179 \\ 0.0364 & -0.0584 \\ 0.0000 & 0.0000 \end{bmatrix}$$

$$\eta = 0.1$$

$$b_{new} = b - \eta \frac{\partial L}{\partial b}$$

$$= \begin{bmatrix} 0.8175 & 0.4613 \\ -0.3599 & -0.4824 \\ 0.1268 & 0.1540 \\ -0.0358 & -0.5677 \end{bmatrix} - 0.1 \cdot \begin{bmatrix} 0.0539 & -0.0865 \\ 0.0111 & -0.0179 \\ 0.0364 & -0.0584 \\ 0.0000 & 0.0000 \end{bmatrix}$$

$$b_{new} = \begin{bmatrix} 0.8121 & 0.4699 \\ -0.3611 & -0.4807 \\ 0.1231 & 0.1598 \\ -0.0358 & -0.5677 \end{bmatrix}$$

### 6.8 Following Epochs Loss Values

$$L_1 = -y_1 \cdot \ln(\hat{y}_1)$$

$$= - \begin{bmatrix} 0. \\ 1. \\ 0. \\ 0. \end{bmatrix} \cdot \ln \left( \begin{bmatrix} 0.2110 \\ 0.2567 \\ 0.2785 \\ 0.2538 \end{bmatrix} \right)$$

$$L_1 = 1.3600$$

$$L_2 = -y_2 \cdot \ln(\hat{y}_2)$$

$$= - \begin{bmatrix} 0. \\ 0. \\ 1. \\ 0. \end{bmatrix} \cdot \ln \left( \begin{bmatrix} 0.2338 \\ 0.2454 \\ 0.2739 \\ 0.2469 \end{bmatrix} \right)$$

$$L_2 = 1.2950$$

$$
\begin{aligned}
L_3 &= -y_3 \cdot \ln(\hat{y}_3) \\
&= -\begin{bmatrix} 0. \\ 0. \\ 1. \\ 0. \end{bmatrix} \cdot \ln\left(\begin{bmatrix} 0.2053 \\ 0.2596 \\ 0.2796 \\ 0.2555 \end{bmatrix}\right) \\
L_3 &= 1.2742
\end{aligned}
$$

$$
\begin{aligned}
L_4 &= -y_4 \cdot \ln(\hat{y}_4) \\
&= -\begin{bmatrix} 0. \\ 0. \\ 0. \\ 1. \end{bmatrix} \cdot \ln\left(\begin{bmatrix} 0.2201 \\ 0.2521 \\ 0.2767 \\ 0.2510 \end{bmatrix}\right) \\
L_4 &= 1.3821
\end{aligned}
$$

$$
\sum_{t=1}^{k} L_t = 1.3600 + 1.2950 + 1.2742 + 1.3821 = 5.3114
$$

Up to Epoch 50 the minimum Total Loss will be 2.9268
Up to Epoch 100 the minimum Total Loss will be 1.7800
Up to Epoch 500 the minimum Total Loss will be 1.0000
Up to Epoch 1000 the minimum Total Loss will be 0.8795.