



Numerical RNN - notes

Deep Neural networks (Birla Institute of Technology and Science, Pilani)



Scan to open on Studocu

Deep Neural Networks

Recurrent Neural Network

Course: AIML ZG511
Instructor: Prof. Seetha Parameswaran
BITS Pilani - Work Integrated Learning Programmes

1 GRU

Problem Setup:

- Input dimension: $d_x = 2$ (vocabulary size)
- Hidden dimension: $d_h = 3$
- Current input: $x_t = \begin{bmatrix} 0.8 \\ 0.6 \end{bmatrix}$
- Previous hidden state: $h_{t-1} = \begin{bmatrix} 0.5 \\ -0.2 \\ 0.3 \end{bmatrix}$

Weight Matrices:

$$\begin{aligned} W_r &= \begin{bmatrix} 0.2 & 0.1 \\ 0.4 & -0.2 \\ -0.3 & 0.1 \end{bmatrix}, & U_r &= \begin{bmatrix} 0.3 & 0.1 & 0.2 \\ -0.1 & 0.4 & 0.2 \\ 0.2 & -0.1 & 0.3 \end{bmatrix} \\ W_z &= \begin{bmatrix} 0.1 & 0.3 \\ -0.2 & 0.1 \\ 0.2 & 0.4 \end{bmatrix}, & U_z &= \begin{bmatrix} 0.4 & -0.1 & 0.2 \\ 0.2 & 0.3 & -0.1 \\ -0.1 & 0.2 & 0.4 \end{bmatrix} \\ W_h &= \begin{bmatrix} 0.3 & -0.1 \\ 0.1 & 0.2 \\ 0.2 & -0.3 \end{bmatrix}, & U_h &= \begin{bmatrix} 0.2 & 0.3 & -0.1 \\ -0.2 & 0.1 & 0.4 \\ 0.3 & -0.2 & 0.1 \end{bmatrix} \end{aligned}$$

Bias vectors: $b_r = b_z = b_h = \begin{bmatrix} 0.1 \\ 0.0 \\ -0.1 \end{bmatrix}$

Step 1: Reset Gate Computation**Reset Gate Formula:** $r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$ **Step 1.1: Compute $W_r x_t$**

$$\begin{aligned}
 W_r x_t &= \begin{bmatrix} 0.2 & 0.1 \\ 0.4 & -0.2 \\ -0.3 & 0.1 \end{bmatrix} \begin{bmatrix} 0.8 \\ 0.6 \end{bmatrix} \\
 &= \begin{bmatrix} 0.2(0.8) + 0.1(0.6) \\ 0.4(0.8) + (-0.2)(0.6) \\ (-0.3)(0.8) + 0.1(0.6) \end{bmatrix} \\
 &= \begin{bmatrix} 0.16 + 0.06 \\ 0.32 - 0.12 \\ -0.24 + 0.06 \end{bmatrix} = \begin{bmatrix} 0.22 \\ 0.20 \\ -0.18 \end{bmatrix}
 \end{aligned}$$

Step 1.2: Compute $U_r h_{t-1}$

$$\begin{aligned}
 U_r h_{t-1} &= \begin{bmatrix} 0.3 & 0.1 & 0.2 \\ -0.1 & 0.4 & 0.2 \\ 0.2 & -0.1 & 0.3 \end{bmatrix} \begin{bmatrix} 0.5 \\ -0.2 \\ 0.3 \end{bmatrix} \\
 &= \begin{bmatrix} 0.15 - 0.02 + 0.06 \\ -0.05 - 0.08 + 0.06 \\ 0.10 + 0.02 + 0.09 \end{bmatrix} = \begin{bmatrix} 0.19 \\ -0.07 \\ 0.21 \end{bmatrix}
 \end{aligned}$$

Step 1.3: Add bias and apply sigmoid

$$\begin{aligned}
 r_t &= \sigma \left(\begin{bmatrix} 0.22 \\ 0.20 \\ -0.18 \end{bmatrix} + \begin{bmatrix} 0.19 \\ -0.07 \\ 0.21 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.0 \\ -0.1 \end{bmatrix} \right) \\
 &= \sigma \left(\begin{bmatrix} 0.51 \\ 0.13 \\ -0.07 \end{bmatrix} \right) = \begin{bmatrix} 0.625 \\ 0.532 \\ 0.483 \end{bmatrix}
 \end{aligned}$$

Step 2: Update Gate Computation**Update Gate Formula:** $z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$ **Step 2.1: Compute $W_z x_t$**

$$\begin{aligned}
 W_z x_t &= \begin{bmatrix} 0.1 & 0.3 \\ -0.2 & 0.1 \\ 0.2 & 0.4 \end{bmatrix} \begin{bmatrix} 0.8 \\ 0.6 \end{bmatrix} \\
 &= \begin{bmatrix} 0.08 + 0.18 \\ -0.16 + 0.06 \\ 0.16 + 0.24 \end{bmatrix} = \begin{bmatrix} 0.26 \\ -0.10 \\ 0.40 \end{bmatrix}
 \end{aligned}$$

Step 2.2: Compute $U_z h_{t-1}$

$$\begin{aligned}
 U_z h_{t-1} &= \begin{bmatrix} 0.4 & -0.1 & 0.2 \\ 0.2 & 0.3 & -0.1 \\ -0.1 & 0.2 & 0.4 \end{bmatrix} \begin{bmatrix} 0.5 \\ -0.2 \\ 0.3 \end{bmatrix} \\
 &= \begin{bmatrix} 0.20 + 0.02 + 0.06 \\ 0.10 - 0.06 - 0.03 \\ -0.05 - 0.04 + 0.12 \end{bmatrix} = \begin{bmatrix} 0.28 \\ 0.01 \\ 0.03 \end{bmatrix}
 \end{aligned}$$

Step 2.3: Add bias and apply sigmoid

$$\begin{aligned}
 z_t &= \sigma \left(\begin{bmatrix} 0.26 \\ -0.10 \\ 0.40 \end{bmatrix} + \begin{bmatrix} 0.28 \\ 0.01 \\ 0.03 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.0 \\ -0.1 \end{bmatrix} \right) \\
 &= \sigma \left(\begin{bmatrix} 0.64 \\ -0.09 \\ 0.33 \end{bmatrix} \right) = \begin{bmatrix} 0.655 \\ 0.478 \\ 0.582 \end{bmatrix}
 \end{aligned}$$

Step 3: Candidate Hidden State Computation**Candidate Formula:** $\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h)$ **Step 3.1: Compute $r_t \odot h_{t-1}$ (element-wise)**

$$\begin{aligned}
 r_t \odot h_{t-1} &= \begin{bmatrix} 0.625 \\ 0.532 \\ 0.483 \end{bmatrix} \odot \begin{bmatrix} 0.5 \\ -0.2 \\ 0.3 \end{bmatrix} \\
 &= \begin{bmatrix} 0.625 \times 0.5 \\ 0.532 \times (-0.2) \\ 0.483 \times 0.3 \end{bmatrix} = \begin{bmatrix} 0.313 \\ -0.106 \\ 0.145 \end{bmatrix}
 \end{aligned}$$

Step 3.2: Compute $W_h x_t$

$$\begin{aligned}
 W_h x_t &= \begin{bmatrix} 0.3 & -0.1 \\ 0.1 & 0.2 \\ 0.2 & -0.3 \end{bmatrix} \begin{bmatrix} 0.8 \\ 0.6 \end{bmatrix} \\
 &= \begin{bmatrix} 0.24 - 0.06 \\ 0.08 + 0.12 \\ 0.16 - 0.18 \end{bmatrix} = \begin{bmatrix} 0.18 \\ 0.20 \\ -0.02 \end{bmatrix}
 \end{aligned}$$

Step 3.3: Compute $U_h(r_t \odot h_{t-1})$

$$\begin{aligned}
 U_h(r_t \odot h_{t-1}) &= \begin{bmatrix} 0.2 & 0.3 & -0.1 \\ -0.2 & 0.1 & 0.4 \\ 0.3 & -0.2 & 0.1 \end{bmatrix} \begin{bmatrix} 0.313 \\ -0.106 \\ 0.145 \end{bmatrix} \\
 &= \begin{bmatrix} 0.063 - 0.032 - 0.015 \\ -0.063 - 0.011 + 0.058 \\ 0.094 + 0.021 + 0.015 \end{bmatrix} = \begin{bmatrix} 0.016 \\ -0.016 \\ 0.130 \end{bmatrix}
 \end{aligned}$$

Step 4: Final Hidden State Computation**Step 4.1: Complete candidate computation**

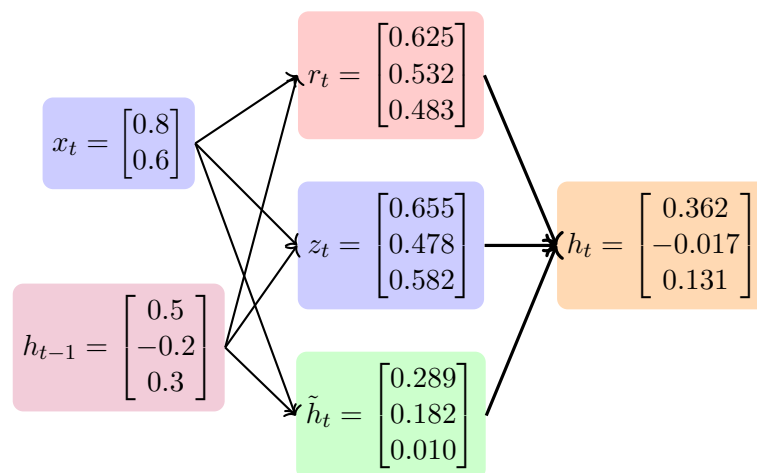
$$\begin{aligned}\tilde{h}_t &= \tanh \left(\begin{bmatrix} 0.18 \\ 0.20 \\ -0.02 \end{bmatrix} + \begin{bmatrix} 0.016 \\ -0.016 \\ 0.130 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.0 \\ -0.1 \end{bmatrix} \right) \\ &= \tanh \left(\begin{bmatrix} 0.296 \\ 0.184 \\ 0.010 \end{bmatrix} \right) = \begin{bmatrix} 0.289 \\ 0.182 \\ 0.010 \end{bmatrix}\end{aligned}$$

Step 4.2: Compute final hidden state GRU Output Formula: $h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$

$$\begin{aligned}(1 - z_t) &= \begin{bmatrix} 1 - 0.655 \\ 1 - 0.478 \\ 1 - 0.582 \end{bmatrix} = \begin{bmatrix} 0.345 \\ 0.522 \\ 0.418 \end{bmatrix} \\ (1 - z_t) \odot h_{t-1} &= \begin{bmatrix} 0.345 \times 0.5 \\ 0.522 \times (-0.2) \\ 0.418 \times 0.3 \end{bmatrix} = \begin{bmatrix} 0.173 \\ -0.104 \\ 0.125 \end{bmatrix} \\ z_t \odot \tilde{h}_t &= \begin{bmatrix} 0.655 \times 0.289 \\ 0.478 \times 0.182 \\ 0.582 \times 0.010 \end{bmatrix} = \begin{bmatrix} 0.189 \\ 0.087 \\ 0.006 \end{bmatrix}\end{aligned}$$

Final Result:

$$h_t = \begin{bmatrix} 0.173 \\ -0.104 \\ 0.125 \end{bmatrix} + \begin{bmatrix} 0.189 \\ 0.087 \\ 0.006 \end{bmatrix} = \begin{bmatrix} \mathbf{0.362} \\ \mathbf{-0.017} \\ \mathbf{0.131} \end{bmatrix}$$

GRU Forward Pass Results

Step 5: Output Layer and Loss Computation**Output Layer Setup:**

- Output dimension: $d_o = 2$ (binary classification)
- Output weight matrix: $W_o = \begin{bmatrix} 0.4 & -0.2 & 0.3 \\ -0.1 & 0.5 & 0.2 \end{bmatrix}$
- Output bias: $b_o = \begin{bmatrix} 0.1 \\ -0.1 \end{bmatrix}$
- True target: $y_{true} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ (one-hot encoded)
- Previous result: $h_t = \begin{bmatrix} 0.362 \\ -0.017 \\ 0.131 \end{bmatrix}$

Step 5.1: Compute logits

$$\begin{aligned}
 z_o &= W_o h_t + b_o \\
 &= \begin{bmatrix} 0.4 & -0.2 & 0.3 \\ -0.1 & 0.5 & 0.2 \end{bmatrix} \begin{bmatrix} 0.362 \\ -0.017 \\ 0.131 \end{bmatrix} + \begin{bmatrix} 0.1 \\ -0.1 \end{bmatrix} \\
 &= \begin{bmatrix} 0.145 + 0.003 + 0.039 \\ -0.036 - 0.009 + 0.026 \end{bmatrix} + \begin{bmatrix} 0.1 \\ -0.1 \end{bmatrix} \\
 &= \begin{bmatrix} 0.187 \\ -0.019 \end{bmatrix} + \begin{bmatrix} 0.1 \\ -0.1 \end{bmatrix} = \begin{bmatrix} 0.287 \\ -0.119 \end{bmatrix}
 \end{aligned}$$

Step 5.2: Apply softmax

$$\begin{aligned}
 \hat{y} &= \text{softmax}(z_o) = \frac{\exp(z_o)}{\sum_i \exp(z_{o,i})} \\
 \exp(z_o) &= \begin{bmatrix} \exp(0.287) \\ \exp(-0.119) \end{bmatrix} = \begin{bmatrix} 1.332 \\ 0.888 \end{bmatrix} \\
 \hat{y} &= \frac{1}{1.332 + 0.888} \begin{bmatrix} 1.332 \\ 0.888 \end{bmatrix} = \frac{1}{2.22} \begin{bmatrix} 1.332 \\ 0.888 \end{bmatrix} = \begin{bmatrix} 0.600 \\ 0.400 \end{bmatrix}
 \end{aligned}$$

Step 5.3: Cross-entropy loss

$$\begin{aligned}
 L &= - \sum_i y_{true,i} \log(\hat{y}_i) \\
 &= -[1 \cdot \log(0.600) + 0 \cdot \log(0.400)] \\
 &= -\log(0.600) = \mathbf{0.511}
 \end{aligned}$$

Step 6: Backpropagation - Output Layer Gradients**Step 6.1: Gradient w.r.t. logits (softmax + cross-entropy)**

$$\frac{\partial L}{\partial z_o} = \hat{y} - y_{true} = \begin{bmatrix} 0.600 \\ 0.400 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.400 \\ 0.400 \end{bmatrix}$$

Step 6.2: Gradient w.r.t. output weights

$$\begin{aligned} \frac{\partial L}{\partial W_o} &= \frac{\partial L}{\partial z_o} h_t^T \\ &= \begin{bmatrix} -0.400 \\ 0.400 \end{bmatrix} \begin{bmatrix} 0.362 & -0.017 & 0.131 \end{bmatrix} \\ &= \begin{bmatrix} -0.145 & 0.007 & -0.052 \\ 0.145 & -0.007 & 0.052 \end{bmatrix} \end{aligned}$$

Step 6.3: Gradient w.r.t. output bias

$$\frac{\partial L}{\partial b_o} = \frac{\partial L}{\partial z_o} = \begin{bmatrix} -0.400 \\ 0.400 \end{bmatrix}$$

Step 6.4: Gradient w.r.t. hidden state (flows to GRU)

$$\begin{aligned} \frac{\partial L}{\partial h_t} &= W_o^T \frac{\partial L}{\partial z_o} \\ &= \begin{bmatrix} 0.4 & -0.1 \\ -0.2 & 0.5 \\ 0.3 & 0.2 \end{bmatrix} \begin{bmatrix} -0.400 \\ 0.400 \end{bmatrix} \\ &= \begin{bmatrix} -0.160 - 0.040 \\ 0.080 + 0.200 \\ -0.120 + 0.080 \end{bmatrix} = \begin{bmatrix} -0.200 \\ 0.280 \\ -0.040 \end{bmatrix} \end{aligned}$$

Step 7: Backpropagation Through GRU - Part 1

Gradient flowing into GRU: $\frac{\partial L}{\partial h_t} = \begin{bmatrix} -0.200 \\ 0.280 \\ -0.040 \end{bmatrix}$

GRU Backward Pass Key Equations: Since $h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$, we need:

Step 7.1: Gradients w.r.t. interpolation components

$$\begin{aligned} \frac{\partial L}{\partial \tilde{h}_t} &= \frac{\partial L}{\partial h_t} \odot z_t \\ &= \begin{bmatrix} -0.200 \\ 0.280 \\ -0.040 \end{bmatrix} \odot \begin{bmatrix} 0.655 \\ 0.478 \\ 0.582 \end{bmatrix} = \begin{bmatrix} -0.131 \\ 0.134 \\ -0.023 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} \frac{\partial L}{\partial h_{t-1}} &= \frac{\partial L}{\partial h_t} \odot (1 - z_t) + \text{gradients from reset gate} \\ &= \begin{bmatrix} -0.200 \\ 0.280 \\ -0.040 \end{bmatrix} \odot \begin{bmatrix} 0.345 \\ 0.522 \\ 0.418 \end{bmatrix} + \Delta h_{reset} \\ &= \begin{bmatrix} -0.069 \\ 0.146 \\ -0.017 \end{bmatrix} + \Delta h_{reset} \text{ (computed below)} \end{aligned}$$

Step 7.2: Gradient w.r.t. update gate

$$\begin{aligned} \frac{\partial L}{\partial z_t} &= \frac{\partial L}{\partial h_t} \odot (\tilde{h}_t - h_{t-1}) \\ &= \begin{bmatrix} -0.200 \\ 0.280 \\ -0.040 \end{bmatrix} \odot \left(\begin{bmatrix} 0.289 \\ 0.182 \\ 0.010 \end{bmatrix} - \begin{bmatrix} 0.5 \\ -0.2 \\ 0.3 \end{bmatrix} \right) \\ &= \begin{bmatrix} -0.200 \\ 0.280 \\ -0.040 \end{bmatrix} \odot \begin{bmatrix} -0.211 \\ 0.382 \\ -0.290 \end{bmatrix} = \begin{bmatrix} 0.042 \\ 0.107 \\ 0.012 \end{bmatrix} \end{aligned}$$

Step 8: Backpropagation Through GRU - Part 2**Step 8.1: Gradient through tanh activation (candidate)**

$$\begin{aligned}
\frac{\partial L}{\partial \tilde{h}_{pre}} &= \frac{\partial L}{\partial \tilde{h}_t} \odot (1 - \tilde{h}_t^2) \\
&= \begin{bmatrix} -0.131 \\ 0.134 \\ -0.023 \end{bmatrix} \odot \left(1 - \begin{bmatrix} 0.289^2 \\ 0.182^2 \\ 0.010^2 \end{bmatrix} \right) \\
&= \begin{bmatrix} -0.131 \\ 0.134 \\ -0.023 \end{bmatrix} \odot \begin{bmatrix} 0.916 \\ 0.967 \\ 1.000 \end{bmatrix} = \begin{bmatrix} -0.120 \\ 0.130 \\ -0.023 \end{bmatrix}
\end{aligned}$$

Step 8.2: Gradients through sigmoid activations For update gate: $\frac{\partial \sigma}{\partial x} = \sigma(x)(1 - \sigma(x))$

$$\begin{aligned}
\frac{\partial L}{\partial z_{pre}} &= \frac{\partial L}{\partial z_t} \odot z_t \odot (1 - z_t) \\
&= \begin{bmatrix} 0.042 \\ 0.107 \\ 0.012 \end{bmatrix} \odot \begin{bmatrix} 0.655 \times 0.345 \\ 0.478 \times 0.522 \\ 0.582 \times 0.418 \end{bmatrix} \\
&= \begin{bmatrix} 0.042 \\ 0.107 \\ 0.012 \end{bmatrix} \odot \begin{bmatrix} 0.226 \\ 0.249 \\ 0.243 \end{bmatrix} = \begin{bmatrix} 0.009 \\ 0.027 \\ 0.003 \end{bmatrix}
\end{aligned}$$

Step 8.3: Reset gate gradient For reset gate, we need to trace gradients through candidate computation:

$$\begin{aligned}
\frac{\partial L}{\partial r_t} &= \frac{\partial L}{\partial \tilde{h}_{pre}} \cdot \frac{\partial \tilde{h}_{pre}}{\partial r_t} \\
&= U_h^T \frac{\partial L}{\partial \tilde{h}_{pre}} \odot h_{t-1} \\
&= \begin{bmatrix} 0.2 & -0.2 & 0.3 \\ 0.3 & 0.1 & -0.2 \\ -0.1 & 0.4 & 0.1 \end{bmatrix} \begin{bmatrix} -0.120 \\ 0.130 \\ -0.023 \end{bmatrix} \odot \begin{bmatrix} 0.5 \\ -0.2 \\ 0.3 \end{bmatrix} \\
&= \begin{bmatrix} -0.024 - 0.026 - 0.007 \\ -0.036 + 0.013 + 0.005 \\ 0.012 + 0.052 - 0.002 \end{bmatrix} \odot \begin{bmatrix} 0.5 \\ -0.2 \\ 0.3 \end{bmatrix} \\
&= \begin{bmatrix} -0.057 \\ -0.018 \\ 0.062 \end{bmatrix} \odot \begin{bmatrix} 0.5 \\ -0.2 \\ 0.3 \end{bmatrix} = \begin{bmatrix} -0.029 \\ 0.004 \\ 0.019 \end{bmatrix}
\end{aligned}$$

Step 9: Reset Gate Backpropagation**Step 9.1: Complete reset gate gradient through sigmoid**

$$\begin{aligned}
\frac{\partial L}{\partial r_{pre}} &= \frac{\partial L}{\partial r_t} \odot r_t \odot (1 - r_t) \\
&= \begin{bmatrix} -0.029 \\ 0.004 \\ 0.019 \end{bmatrix} \odot \begin{bmatrix} 0.625 \times 0.375 \\ 0.532 \times 0.468 \\ 0.483 \times 0.517 \end{bmatrix} \\
&= \begin{bmatrix} -0.029 \\ 0.004 \\ 0.019 \end{bmatrix} \odot \begin{bmatrix} 0.234 \\ 0.249 \\ 0.250 \end{bmatrix} = \begin{bmatrix} -0.007 \\ 0.001 \\ 0.005 \end{bmatrix}
\end{aligned}$$

Step 9.2: Complete gradient w.r.t. h_{t-1} Including gradients from reset gate path:

$$\begin{aligned}
\Delta h_{reset} &= U_r^T \frac{\partial L}{\partial r_{pre}} \\
&= \begin{bmatrix} 0.3 & -0.1 & 0.2 \\ 0.1 & 0.4 & -0.1 \\ 0.2 & 0.2 & 0.3 \end{bmatrix} \begin{bmatrix} -0.007 \\ 0.001 \\ 0.005 \end{bmatrix} \\
&= \begin{bmatrix} -0.002 - 0.0001 + 0.001 \\ -0.0007 + 0.0004 - 0.0005 \\ -0.0014 + 0.0002 + 0.0015 \end{bmatrix} = \begin{bmatrix} -0.001 \\ -0.0008 \\ 0.0003 \end{bmatrix} \\
\\
\frac{\partial L}{\partial h_{t-1}} &= \begin{bmatrix} -0.069 \\ 0.146 \\ -0.017 \end{bmatrix} + \begin{bmatrix} -0.001 \\ -0.0008 \\ 0.0003 \end{bmatrix} = \begin{bmatrix} -0.070 \\ 0.145 \\ -0.017 \end{bmatrix}
\end{aligned}$$

Step 10: Weight Gradient Computation**Step 10.1: Update gate weight gradients**

$$\begin{aligned}
\frac{\partial L}{\partial W_z} &= \frac{\partial L}{\partial z_{pre}} x_t^T \\
&= \begin{bmatrix} 0.009 \\ 0.027 \\ 0.003 \end{bmatrix} \begin{bmatrix} 0.8 & 0.6 \end{bmatrix} \\
&= \begin{bmatrix} 0.007 & 0.005 \\ 0.022 & 0.016 \\ 0.002 & 0.002 \end{bmatrix} \quad (3 \times 2 \text{ matrix})
\end{aligned}$$

$$\begin{aligned}
\frac{\partial L}{\partial U_z} &= \frac{\partial L}{\partial z_{pre}} h_{t-1}^T \\
&= \begin{bmatrix} 0.009 \\ 0.027 \\ 0.003 \end{bmatrix} \begin{bmatrix} 0.5 & -0.2 & 0.3 \end{bmatrix} \\
&= \begin{bmatrix} 0.005 & -0.002 & 0.003 \\ 0.014 & -0.005 & 0.008 \\ 0.002 & -0.001 & 0.001 \end{bmatrix} \quad (3 \times 3 \text{ matrix})
\end{aligned}$$

Step 10.2: Reset gate weight gradients

$$\begin{aligned}
\frac{\partial L}{\partial W_r} &= \frac{\partial L}{\partial r_{pre}} x_t^T \\
&= \begin{bmatrix} -0.007 \\ 0.001 \\ 0.005 \end{bmatrix} \begin{bmatrix} 0.8 & 0.6 \end{bmatrix} \\
&= \begin{bmatrix} -0.006 & -0.004 \\ 0.001 & 0.001 \\ 0.004 & 0.003 \end{bmatrix} \quad (3 \times 2 \text{ matrix})
\end{aligned}$$

$$\begin{aligned}
\frac{\partial L}{\partial U_r} &= \frac{\partial L}{\partial r_{pre}} h_{t-1}^T \\
&= \begin{bmatrix} -0.007 \\ 0.001 \\ 0.005 \end{bmatrix} \begin{bmatrix} 0.5 & -0.2 & 0.3 \end{bmatrix} \\
&= \begin{bmatrix} -0.004 & 0.001 & -0.002 \\ 0.001 & -0.0002 & 0.0003 \\ 0.003 & -0.001 & 0.002 \end{bmatrix} \quad (3 \times 3 \text{ matrix})
\end{aligned}$$

Step 11: Candidate Weight Gradients**Step 11.1: Candidate weight gradients**

$$\begin{aligned}
\frac{\partial L}{\partial W_h} &= \frac{\partial L}{\partial \tilde{h}_{pre}} x_t^T \\
&= \begin{bmatrix} -0.120 \\ 0.130 \\ -0.023 \end{bmatrix} \begin{bmatrix} 0.8 & 0.6 \end{bmatrix} \\
&= \begin{bmatrix} -0.096 & -0.072 \\ 0.104 & 0.078 \\ -0.018 & -0.014 \end{bmatrix} \quad (3 \times 2 \text{ matrix})
\end{aligned}$$

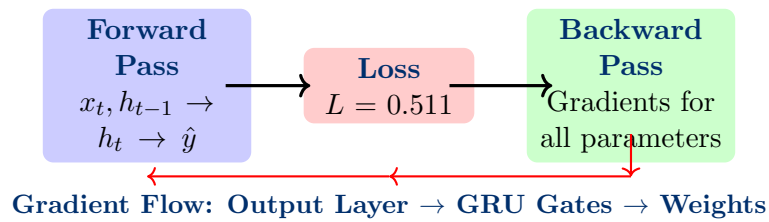
For U_h , we need gradient w.r.t. $(r_t \odot h_{t-1})$:

$$\begin{aligned}
\frac{\partial L}{\partial U_h} &= \frac{\partial L}{\partial \tilde{h}_{pre}} (r_t \odot h_{t-1})^T \\
&= \begin{bmatrix} -0.120 \\ 0.130 \\ -0.023 \end{bmatrix} \begin{bmatrix} 0.313 & -0.106 & 0.145 \end{bmatrix} \\
&= \begin{bmatrix} -0.038 & 0.013 & -0.017 \\ 0.041 & -0.014 & 0.019 \\ -0.007 & 0.002 & -0.003 \end{bmatrix} \quad (3 \times 3 \text{ matrix})
\end{aligned}$$

Step 11.2: Bias gradients All bias gradients equal their respective pre-activation gradients:

$$\begin{aligned}
\frac{\partial L}{\partial b_z} &= \frac{\partial L}{\partial z_{pre}} = \begin{bmatrix} 0.009 \\ 0.027 \\ 0.003 \end{bmatrix} \\
\frac{\partial L}{\partial b_r} &= \frac{\partial L}{\partial r_{pre}} = \begin{bmatrix} -0.007 \\ 0.001 \\ 0.005 \end{bmatrix} \\
\frac{\partial L}{\partial b_h} &= \frac{\partial L}{\partial \tilde{h}_{pre}} = \begin{bmatrix} -0.120 \\ 0.130 \\ -0.023 \end{bmatrix}
\end{aligned}$$

Summary: Complete GRU Training Step with Corrected Dimensions



Gradient Summary:

Parameter	Dimensions	Max Gradient	Comments
W_o, b_o	$(2 \times 3), (2 \times 1)$	0.400	Output layer (largest)
W_h	(3×2)	0.104	Candidate weights
U_h	(3×3)	0.041	Candidate recurrent
W_z	(3×2)	0.022	Update gate weights
U_z	(3×3)	0.014	Update recurrent
W_r	(3×2)	0.006	Reset gate weights
U_r	(3×3)	0.004	Reset recurrent

Parameter Updates: With learning rate $\eta = 0.01$, update weights simultaneously:

$$\theta_{new} = \theta_{old} - 0.01 \times \frac{\partial L}{\partial \theta}$$