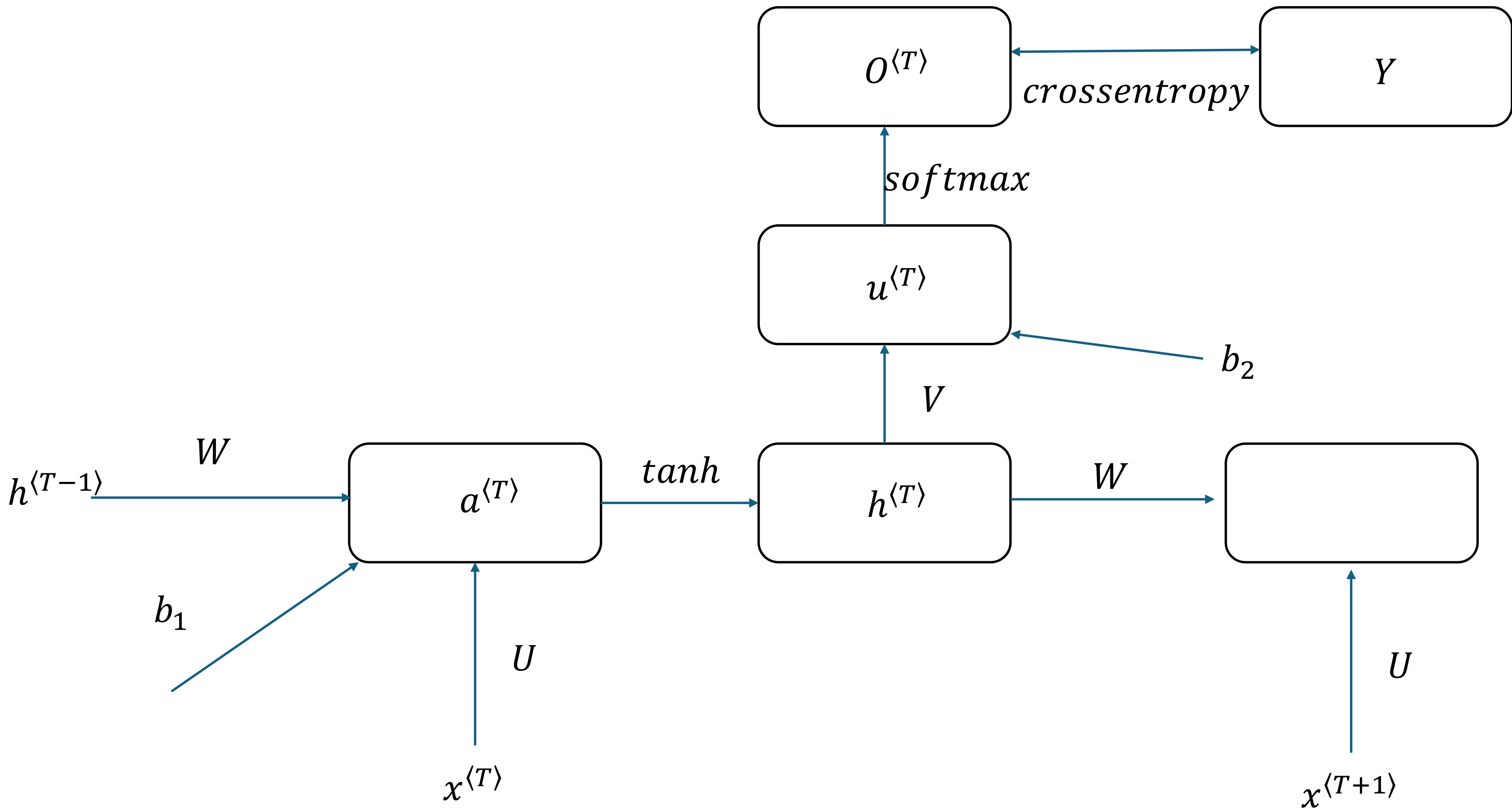
Variables and functions we need

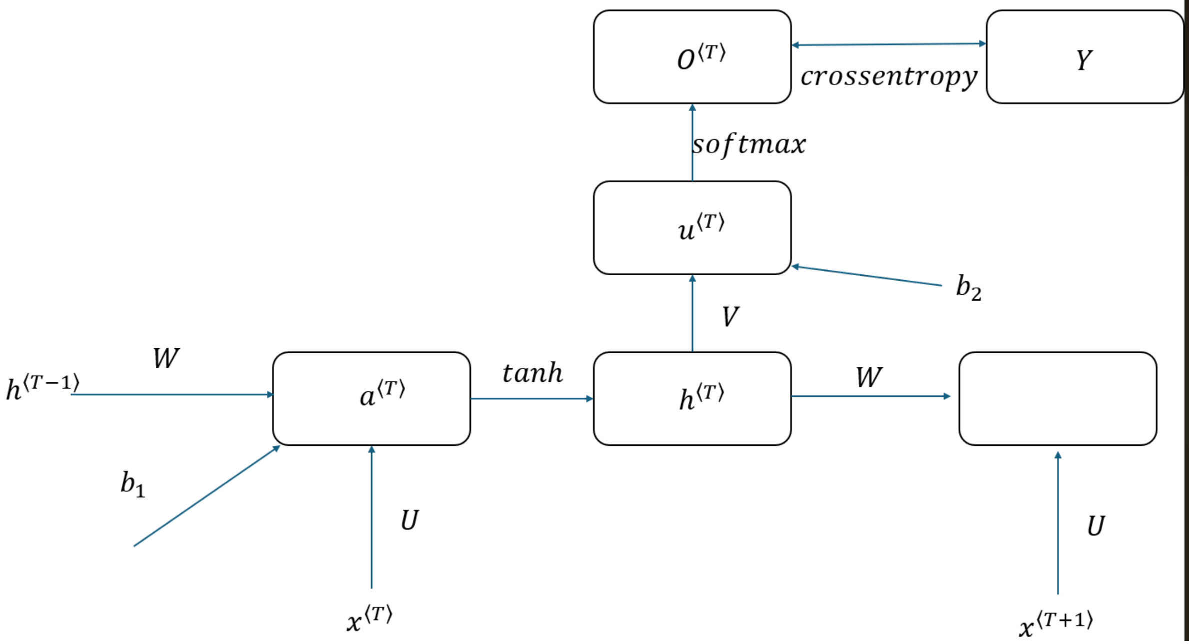
- Input_hidden weights
- Hidden_hidden weights
- Bias1
- Tanh
- Hidden_output weights
- Bias2
- Softmax：$(\frac{x_0}{\sum e^{x_i}}, \cdots, \frac{x_9}{\sum e^{x_i}})$
- Cross entropy：$-\frac{1}{N}\sum_{n=0}^{N-1}\sum_{i=0}^{9}(yt_i^{(n)}\log(yp_i^{(n)}))$



$O^{\langle T \rangle}$
$crossentropy$
Y
$softmax$
$u^{\langle T \rangle}$
b_2
V
W
$h^{\langle T-1 \rangle}$
$a^{\langle T \rangle}$
$tanh$
$h^{\langle T \rangle}$
W
b_1
U
U
$x^{\langle T \rangle}$
$x^{\langle T+1 \rangle}$

Forward

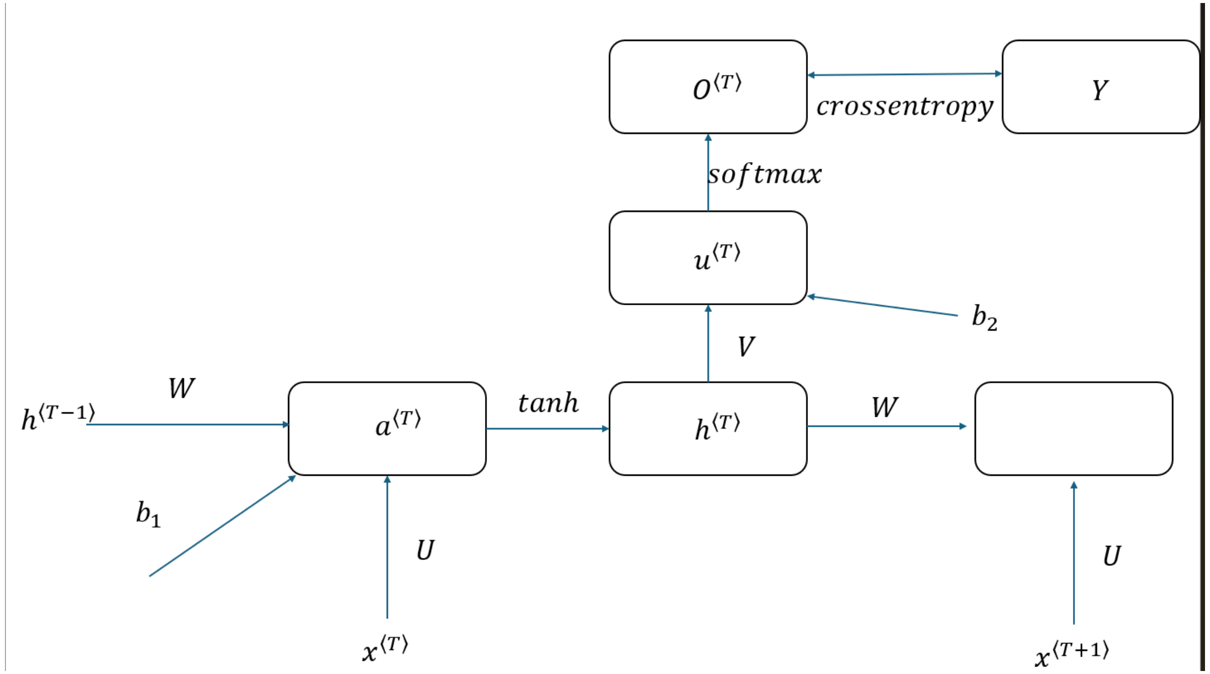
- $a^{\langle T\rangle} = U * x^{\langle T\rangle} + W * h^{\langle T-1\rangle} + \ b_1$
- $h^{\langle T\rangle} = \tanh(a^{\langle T\rangle})$ (hidden state)
- $u^{\langle T\rangle} = V * h^{\langle T\rangle} + \ b_2$
- $O^{\langle T\rangle} = softmax(u^{\langle T\rangle})$ (output at time T)
- $Loss = -\sum_{i=0}^{9} Y_i * \log(O_i^{\langle T\rangle})$



- In this case, we have 28*28 images with 10 categories
- Time steps：28
- Input： $x_{28\times 1}$
- Hidden state：$h_{256\times 1}$
- Biases：$b1_{256\times 1}, b2_{10\times 1}$
- Weights：$U_{256\times 28}, W_{256\times 256}, V_{10\times 256}$

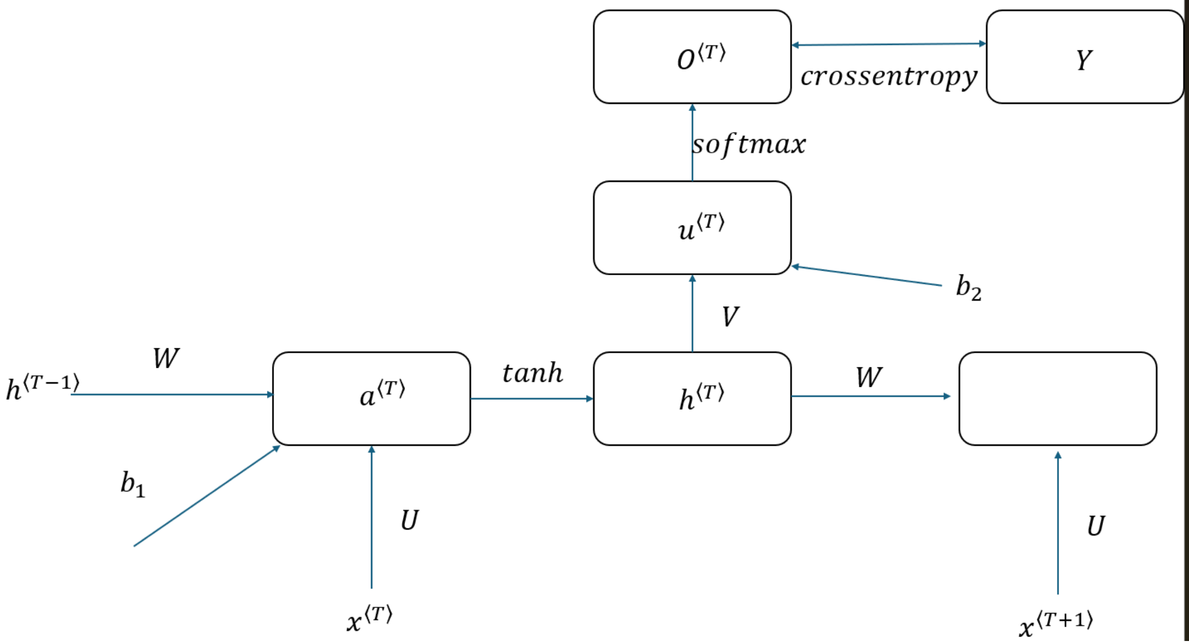
Back propagation ： gradient descent

- Want to know：
- $\frac{\partial Loss}{\partial V}$, $\frac{\partial Loss}{\partial b_2}$, $\frac{\partial Loss}{\partial U}$, $\frac{\partial Loss}{\partial W}$, $\frac{\partial Loss}{\partial b_1}$



Forward

- $a^{\langle T \rangle} = U * x^{\langle T \rangle} + W * h^{\langle T-1 \rangle} + \ b_1$
- $h^{\langle T \rangle} = \tanh(a^{\langle T \rangle})$ (hidden state)
- $u^{\langle T \rangle} = V * h^{\langle T \rangle} + \ b_2$
- $O^{\langle T \rangle} = softmax(u^{\langle T \rangle})$ (output at time T)
- $Loss = -\sum_{i=0}^{9} Y_i * \log(O_i^{\langle T \rangle})$



$$\frac{\partial Loss}{\partial V}$$

- For example, $\frac{\partial Loss}{\partial v_{12}}$
- $\begin{pmatrix} v_{00} & v_{01} & v_{02} & & \\ v_{10} & v_{11} & v_{12} & \cdots & \\ v_{20} & v_{21} & v_{22} & & \\ & \vdots & & \ddots & \vdots \\ & & & \cdots & \end{pmatrix} * \begin{matrix} h_0 \\ h_1 \\ h_2 \\ \vdots \end{matrix} + \begin{matrix} b_0 \\ \vdots \\ b_9 \end{matrix} = \begin{matrix} u_0 \\ u_1 \\ \vdots \\ u_9 \end{matrix}$

- $\frac{\partial Loss}{\partial v_{12}} = \frac{\partial Loss}{\partial u_1} * \frac{\partial u_1}{\partial v_{12}}$
- $\frac{\partial u_1}{\partial v_{12}} = h_2$ and $\frac{\partial Loss}{\partial u_1} = \sum_{i=0}^{9} \frac{\partial Loss}{\partial O_i} * \frac{\partial O_i}{\partial u_1}$

$$\frac{\partial Loss}{\partial u_1} = \sum_{i=0}^{9} \frac{\partial Loss}{\partial O_i} * \frac{\partial O_i}{\partial u_1}$$

- $\frac{\partial Loss}{\partial O_i} = -\frac{Y_i}{O_i}$
- For i = 1, $\frac{\partial O_i}{\partial u_1} = O_1(1 - O_1)$
- For i $\neq$ 1, $\frac{\partial O_i}{\partial u_1} = -O_i * O_1$
- $\frac{\partial Loss}{\partial u_1} = -Y_1 + O_1 * (\sum_{i=0}^{9} Y_i)$
- Since we use one-hot code $\sum_{i=0}^{9} Y_i = 1$
- Hence, $\frac{\partial Loss}{\partial u_1} = O_1 - Y_1$
- $\frac{\partial Loss}{\partial v_{12}} = (O_1 - Y_1) * h_2$, that is, $\frac{\partial Loss}{\partial v_{ij}} = \frac{1}{N} * \sum_{n=0}^{N-1} \left({O^{\langle T \rangle}}_i^n - Y_i^n \right) * {h^{\langle T \rangle}}_j^n$

Similar to $\frac{\partial Loss}{\partial b2_1} = \sum_{i=0}^{9} \frac{\partial Loss}{\partial O_i} * \frac{\partial O_i}{\partial b2_1}$

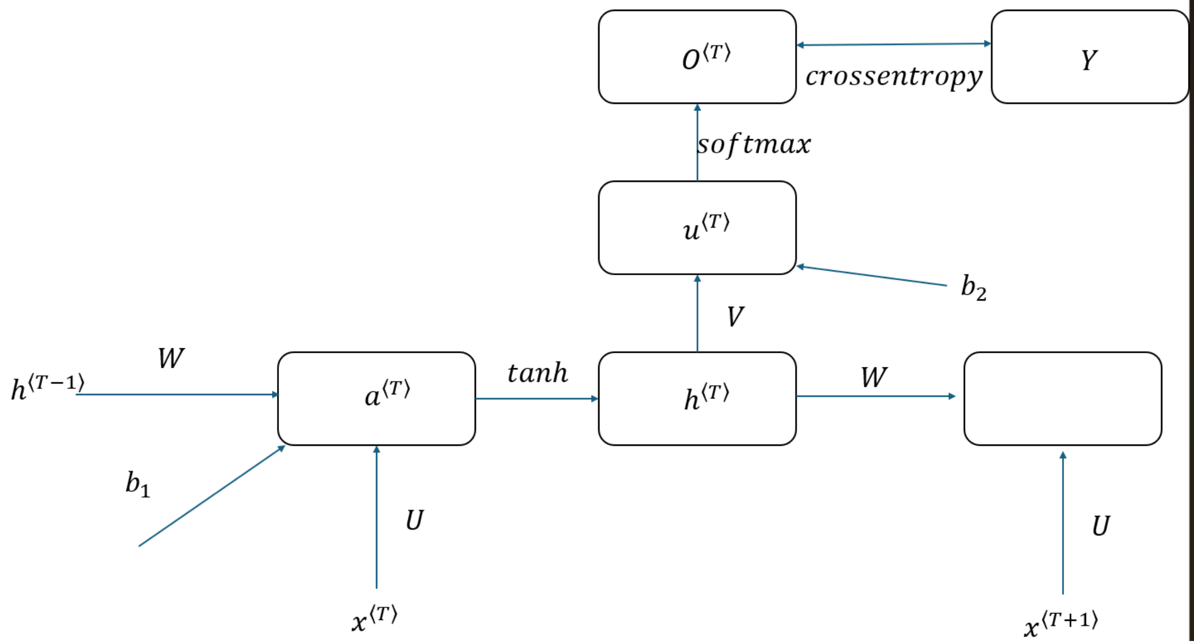
- $\frac{\partial Loss}{\partial b2_1} = (O_1 - Y_1)$, that is, $\frac{\partial Loss}{\partial b2_i} = \frac{1}{N} * \sum_{n=0}^{N-1} \left({O^{\langle T \rangle}}_i^n - Y_i^n \right)$

- Note that both of $\frac{\partial Loss}{\partial V}$, $\frac{\partial Loss}{\partial b_2}$ and following terms have $\left({O^{\langle T\rangle}}_i^n - Y_i^n\right)$
- Hence, in programming we let delta as $\left({O^{\langle T\rangle}}^n - Y^n\right)$

- Now we compute $\frac{\partial Loss}{\partial U}$, $\frac{\partial Loss}{\partial W}$, $\frac{\partial Loss}{\partial b_1}$

Forward

- $a^{\langle T \rangle} = U * x^{\langle T \rangle} + W * h^{\langle T-1 \rangle} + \ b_1$
- $h^{\langle T \rangle} = \tanh(a^{\langle T \rangle})$ (hidden state)
- $u^{\langle T \rangle} = V * h^{\langle T \rangle} + \ b_2$
- $O^{\langle T \rangle} = softmax(u^{\langle T \rangle})$ (output at time T)
- $Loss = -\sum_{i=0}^{9} Y_i * \log(O_i^{\langle T \rangle})$



$$\frac{\partial Loss}{\partial U}$$

- For example, $\frac{\partial Loss}{\partial u_{12}}$
- $\begin{pmatrix} u_{11} & u_{12} & u_{13} & & \\ u_{21} & u_{22} & u_{23} & \cdots & \\ u_{31} & u_{32} & u_{33} & & \\ & \vdots & & \ddots & \vdots \\ & & & \cdots & \end{pmatrix} * \begin{matrix} x_1 \\ x_2 \\ x_3 \\ \vdots \end{matrix} + \begin{pmatrix} w_{11} & w_{12} & w_{13} & & \\ w_{21} & w_{22} & w_{23} & \cdots & \\ w_{31} & w_{32} & w_{33} & & \\ & \vdots & & \ddots & \vdots \\ & & & \cdots & \end{pmatrix} * \begin{matrix} h_1 \\ h_2 \\ \vdots \end{matrix} + \begin{matrix} b_1 \\ \vdots \end{matrix} = \begin{matrix} a_1 \\ a_2 \\ \vdots \end{matrix}$
- $\frac{\partial Loss}{\partial u_{12}} = \frac{\partial Loss}{\partial h_0} \frac{\partial h_0}{\partial a_1} \frac{\partial a_1}{\partial u_{12}}$
- Note that, at time T, $\frac{\partial a_1}{\partial u_{12}} = \frac{\partial a_1}{\partial u_{12}} + \frac{\partial a_1}{\partial h^{\langle T-1 \rangle}} \frac{\partial h^{\langle T-1 \rangle}}{\partial u_{12}} = x_2 + (\sum_i w_{1i} \frac{\partial h_i^{\langle T-1 \rangle}}{\partial u_{12}})$

$$\frac{\partial Loss}{\partial h_0}\frac{\partial h_0}{\partial a_1}$$

- $\frac{\partial Loss}{\partial h_0} = \sum_i \frac{\partial Loss}{\partial u_i}\frac{\partial u_i}{\partial h_0} = \sum_i (O_i - Y_i)(v_{i0})$, (note $u_j = \sum_i v_{ji} h_i + b_j$)
- By $h^{\langle T \rangle} = \tanh(a^{\langle T \rangle})$, $\frac{\partial h_0}{\partial a_1} = 1 - (\tanh(a_1))^2$
- Hence

$$\frac{\partial Loss}{\partial u_{12}} = \frac{\partial Loss}{\partial h_0}\frac{\partial h_0}{\partial a_1}\frac{\partial a_1}{\partial u_{12}}$$

$$= (\sum_i (O_i - Y_i)(v_{i0})) * (1 - (\tanh(a_1))^2) * (x_2 + \left(\sum_j w_{1j} \frac{\partial h_j^{\langle T-1 \rangle}}{\partial u_{12}}\right))$$

- $\frac{\partial Loss}{\partial u_{ij}} = \; (\sum_k (O_k - Y_k)(v_{ki}))(1 - (\tanh(a_i))^2) * (x_j + (\sum_l w_{il} \frac{\partial h_l^{\langle T-1 \rangle}}{\partial u_{ij}}))$

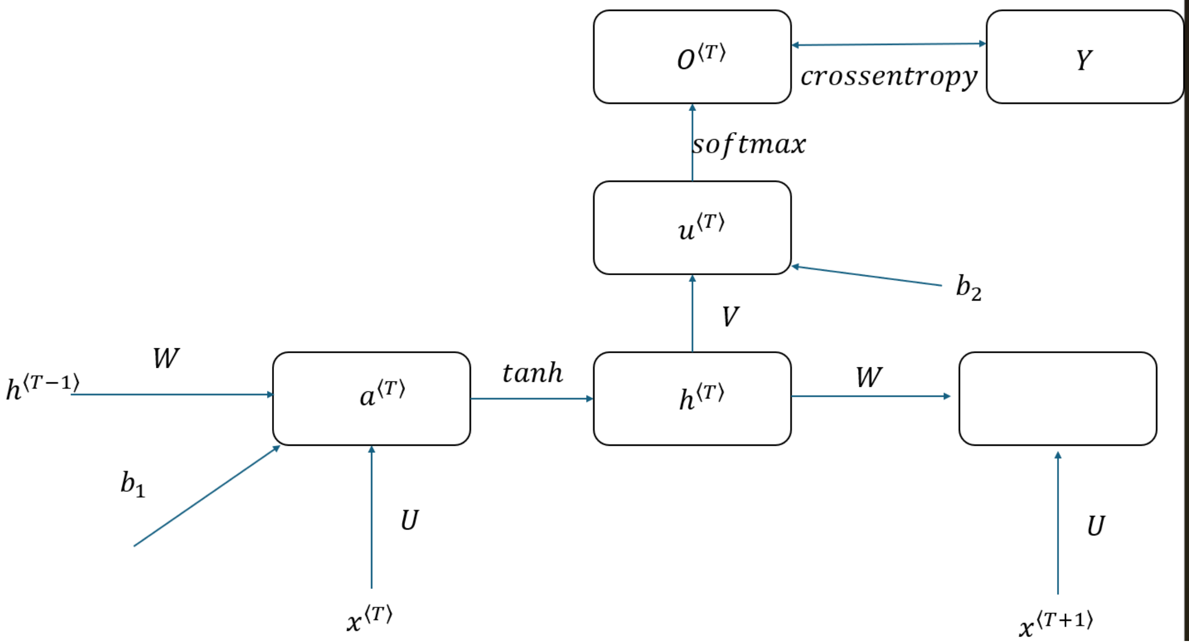
- Formally, write $\frac{\partial Loss}{\partial U} = delta * V * \left(1 - {h^{\langle T \rangle}}^2\right) * \left(x^{\langle T \rangle} + W * \frac{\partial h^{\langle T-1 \rangle}}{\partial U}\right)$

- Similar for $\frac{\partial Loss}{\partial W}$ and $\frac{\partial Loss}{\partial b_1}$

- $\frac{\partial Loss}{\partial W} = delta * V * \left(1 - h^{\langle T \rangle^2}\right) * \left(h^{\langle T-1 \rangle} + W * \frac{\partial h^{\langle T-1 \rangle}}{\partial W}\right)$
- $\frac{\partial Loss}{\partial b_1} = delta * V * \left(1 - h^{\langle T \rangle^2}\right) * \left(1 + W * \frac{\partial h^{\langle T-1 \rangle}}{\partial b_1}\right)$

Forward

- $a^{\langle T \rangle} = U * x^{\langle T \rangle} + W * h^{\langle T-1 \rangle} + \ b_1$
- $h^{\langle T \rangle} = \tanh(a^{\langle T \rangle})$ (hidden state)
- $u^{\langle T \rangle} = V * h^{\langle T \rangle} + \ b_2$
- $O^{\langle T \rangle} = softmax(u^{\langle T \rangle})$ (output at time T)
- $Loss = -\sum_{i=0}^{9} Y_i * \log(O_i^{\langle T \rangle})$



$$\frac{\partial h^{\langle T-1\rangle}}{\partial U}, \frac{\partial h^{\langle T-1\rangle}}{\partial W}, \frac{\partial h^{\langle T-1\rangle}}{\partial b_1}$$

- By previous calculations,
- $\frac{\partial h^{\langle T-1\rangle}}{\partial U} = \left(1 - h^{\langle T-1\rangle^2}\right) * \left(x^{\langle T\rangle} + W * \frac{\partial h^{\langle T-2\rangle}}{\partial U}\right)$
- $\frac{\partial h^{\langle T-1\rangle}}{\partial W} = \left(1 - h^{\langle T-1\rangle^2}\right) * \left(h^{\langle T-2\rangle} + W * \frac{\partial h^{\langle T-2\rangle}}{\partial W}\right)$
- $\frac{\partial h^{\langle T-1\rangle}}{\partial b_1} = \left(1 - h^{\langle T-1\rangle^2}\right) * \left(1 + W * \frac{\partial h^{\langle T-2\rangle}}{\partial b_1}\right)$
- Hence, we calculate $\frac{\partial Loss}{\partial U}, \frac{\partial Loss}{\partial W}, \frac{\partial Loss}{\partial b_1}$ recursively.
- Every recursive steps we need multiply W and $\left(1 - h^{\langle T-1\rangle^2}\right)$

Data preparation

- We need extra dimension for time step
- Three dimensions are (Time, Batch size, node/data)