Vanilla RNN

$$h_t = \tanh\left(W\begin{pmatrix}h_{t-1}\\ x_t\end{pmatrix}\right)$$

Vanilla LSTM

$$\begin{pmatrix}hi\\ hf\\ ho\\ hc\end{pmatrix} = \begin{pmatrix}\sigma\\ \sigma\\ \sigma\\ \tanh(\cdot)\end{pmatrix}\begin{pmatrix}W_i\\ W_f\\ W_o\\ W_c\end{pmatrix}\begin{pmatrix}h_{t-1}\\ x_t\end{pmatrix}$$

$$c_t = hf \odot c_{t-1} + hi \odot hc$$

$$h_t = ho \odot \tanh(c_t)$$

c_t, cell state, hidden from the outside world
hf, forget gate, indicator to keep or forget previous cell state element
hi, input gate, indicator to edit cell state element
hc, cell gate, increment or decrement cell state element by a value between -1 and 1
ho, output gate, indicator to reveal previous/edited cell state element to outside world

$$h_t = (1, H) = number\ of\ LSTM\ layer\ neurons$$

$$x_t = (1, D) = number\ of\ items\ in\ vocabulary$$

$$\hat{y}_t = (1, D) = output\ is\ probabilities\ over\ vocabulary\ set$$

$$Z = H + D = concatenated\ size$$

$$W_f, W_i, W_o, W_c = (Z, H) \quad b_f, b_i, b_o, b_c = (1, H) \quad hf_t, hi_t, ho_t, hc_t = (1, H)$$

$$W_v = (H, D) \quad b_v = (1, D)$$

$$L_k = -\sum_{t=k}^{T}\sum_{j} y_{t,j} \log \hat{y}_{t,j}$$

$$L = L_1$$

$z_t = [h_{t-1}, x_t]$ $hf_t = \sigma(z_t \cdot W_f + b_f)$ $hi_t = \sigma(z_t \cdot W_i + b_i)$ $ho_t = \sigma(z_t \cdot W_o + b_o)$ $hc_t = \tanh(z_t \cdot W_c + b_c)$ $c_t = hf_t \odot c_{t-1} + hi_t \odot hc_t$ $h_t = ho_t \odot \tanh(c_t)$ $v_t = h_t \cdot W_v + b_v$ $\hat{y}_t = \text{softmax}(v_t)$	$dv_t = \hat{y}_t - y_t$ $dh_t = dv_t \cdot W_v^\top + dh_t'$ $dho_t = dh_t \odot \tanh(c_t)$ $dc_t = dh_t \odot o_t \odot \left(1 - \tanh^2(c_t)\right) + dc_t'$ $dhf_t = dc_t \odot c_{t-1}$ $dhi_t = dc_t \odot hc_t$ $dhc_t = dc_t \odot hi_t$ $dhf_t' = hf_t \odot (1 - hf_t) \odot dhf_t$ $dhi_t' = hi_t \odot (1 - hi_t) \odot dhi_t$ $dhc_t' = (1 - hc_t^2) \odot dhc_t$ $dho_t' = ho_t \odot (1 - ho_t) \odot dho_t$ $dz_t = dhf_t' \cdot W_f^\top + dhi_t' \cdot W_i^\top + dho_t' \cdot W_o^\top + dhc_t' \cdot W_c^\top$ $dW_v = h_t^\top \cdot dv_t \quad db_v = dv_t$ $dW_f = z^\top \cdot dhf_t' \quad db_f = dhf_t'$ $dW_i = z^\top \cdot dhi_t' \quad db_i = dhi_t'$ $dW_c = z^\top \cdot dhc_t' \quad db_c = dhc_t'$ $dW_o = z^\top \cdot dho_t' \quad db_o = dho_t'$ $[dh_{t-1}', dx_t] = dz_t$ $dc_t' = hf_t \odot dc_t$

yhat = softmax(v)	dv = yhat.copy() dv[1, y_train] -= 1

v = h @ Wv + bv	dh = dv @ Wv.T #(1,H)=(1,D)(D,H) dh += dh_next dWv = h.T @ dv #(H,D)=(H,1)(1,D) dbv = dv * 1
h = ho * tanh(c)	dho = dh * tanh(c) dc = dh * ho * dtanh(c) dc += dc_next
c = hf * c_prev + hi * hc	dhf = dc * c_prev dhi = dc * hc dhc = dc * hi
hf = sigmoid(z @ Wf + bf)	dhf *= dsigmoid(hf) dz = dhf @ Wf.T #(1,Z)=(1,H)(H,Z) dWf = z.T @ dhf #(Z,H)=(Z,1)(1,H) dbf = dhf * 1 #(1,H)=(1,H)
hi = sigmoid(z @ Wi + bi)	dhi *= dsigmoid(hi) dz += dhi @ Wi.T #(1,Z)=(1,H)(H,Z) dWi = z.T @ dhi #(Z,H)=(Z,1)(1,H) dbi = dhi * 1 #(1,H)=(1,H)
ho = sigmoid(z @ Wo + bo)	dho *= dsigmoid(ho) dz += dho @ Wo.T #(1,Z)=(1,H)(H,Z) dWo = z.T @ dho #(Z,H)=(Z,1)(1,H) dbo = dho * 1 #(1,H)=(1,H)

hc = tanh(z @ Wc + bc)	dhc *= dtanh(hc)
	dz += dhc @ Wc.T #(1,Z)=(1,H)(H,Z)
	dWc = z.T @ dhc #(Z,H)=(Z,1)(1,H)
	dbc = dhc * 1 #(1,H)=(1,H)
	
	dh_next = dz[:, 1:H] #Z=H+D
	dc_next = hf * dc