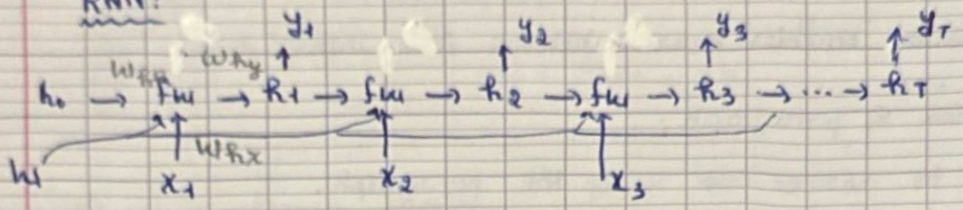
$$\frac{e^{s}+e^{-s}}{e^{s}-e^{-s}} = \frac{e^{s}+\frac{1}{e^{s}}}{e^{s}-\frac{1}{e^{s}}} = \frac{\frac{e^{2s}+1}{e^{s}}}{\frac{e^{2s}-1}{e^{s}}} = \frac{e^{2s}+1}{e^{2s}-1}$$

HW 5: Image Captioning.

RNN:



$$h_t = f_W(h_{t-1}, x_t)$$
$$= \theta(W_{hh} h_{t-1} + W_{hx} x_t)$$
$$= \theta\left([W_{hh} \; W_{hx}] \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} + b_h\right)$$

Many to Many:
$$y_t = \theta(W_{hy} h_t + b_y)$$

ENCODER: Encode input sequence into single vector;
MANY to ONE; $y = W_{hy} h_T + b_y$

LSTM:

$$\begin{bmatrix} i \\ f \\ o \\ g \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} W \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$
$$h_t = o \odot \tanh(c_t).$$

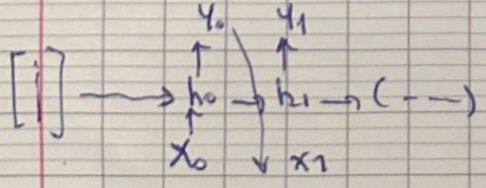
Image captioning:

Encoder: CNN to transform image into vector.

Decoder: RNN.

Image → [ ]

x_0 = [START]



$$h_t = \tanh(W_{hx} x_t + W_{hh} h_{t-1} + b_h).$$
$$y_t = W_{hy} h_t + b_y.$$
$$L = \frac{1}{2} \sum_t (y_t - \hat{y}_t)^2$$

EX:
$T = 2$; $x = [1, 2]$; $W_{hx} = \begin{bmatrix} 0.5 \\ -0.3 \end{bmatrix}$; $W_{hh} = \begin{bmatrix} 0.1 & 0.2 \\ 0.4 & 0.5 \end{bmatrix}$
$W_{hy} = [0.6 \;\; 0.4]$; $b_h = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$; $b_y = 0$

Fwd pass: $t=1$:
$$h_1 = \tanh\left(\begin{bmatrix} 0.5 \\ -0.3 \end{bmatrix} \cdot 1 + \begin{bmatrix} 0 \\ 0 \end{bmatrix}\right) = \begin{pmatrix} 0.462 \\ -0.291 \end{pmatrix}.$$
$$y_1 = [0.6 \;\; 0.4](h_1) = 0.277$$

$t=2$:
$$h_2 = \tanh\left(\begin{pmatrix} 0.5 \\ -0.3 \end{pmatrix} \cdot 2 + \begin{pmatrix} 0.1 & 0.2 \\ 0.4 & 0.5 \end{pmatrix} \begin{pmatrix} 0.462 \\ -0.291 \end{pmatrix}\right) = \begin{pmatrix} 0.744 \\ -0.547 \end{pmatrix}$$

$y_2 = 0.246$. / $y_{act} = [1.5 \;\; 2.0]$.

$L = 1.423$.

$$\frac{\partial L}{\partial W_{hy}} = \sum_t \frac{\partial L}{\partial y_t} \cdot \frac{\partial y_t}{\partial W_{hy}}$$

$$\frac{\partial L}{\partial y_2} = y_2 - \hat{y}_2 = -1.754.$$

$$\frac{\partial y_2}{\partial W_{hy}} = h_2$$

$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

$\frac{\partial \tanh}{\partial g} =$

HW 5:

train 2014 - Vgg16 - pca
val 2014 - Vgg16 - pca
} Extracted Features & PCA applied.

caption - encoding. Integer ID → word.

model_layers.py: contains all fundamental RNN layer implem.
rnn_model.py: bring layers together, & create Image captioning.

forward_rnn_step: process incoming data & updates internal rnn memory

① Forward layer: fwd pass through linear dense.
(1) inp: Input data
weights matrix.
Bias Vector.
(2) output = inp. weights + bias.

$\begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \end{bmatrix} \cdot \begin{bmatrix} w_1 & w_2 \\ w_3 & w_4 \end{bmatrix}$

② Backward layer:
inp, weights, bias = cache_data

$\partial inp = \nabla \cdot W^T$ ⇒ Propagates error backward

$\frac{\partial L}{\partial W_{hx}} = \frac{\partial L}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial h_t} \times \frac{\partial h_t}{\partial W_{hx}}$

$z_{hx} = W_{hx} x_t + b_h$.
$z_{hh} = W_{hh} h_{t-1}$.
} $z = W_{hx} x_t + W_{hh} h_{t-1} + b_h$.
$h_t = \tanh(z)$.

Given:
d_inp = grd. weights.
d_weights = inp^T grd
d_bias = Σ grd.
} $\partial h_t / \partial z = 1 - \tanh^2(z)$. $\frac{\partial L}{\partial z} = \frac{\partial L}{\partial h_t} \times \frac{\partial h_t}{\partial z}$
We've given $\frac{\partial L}{\partial h_t}$.

$z = W_{hx} x_t + W_{hh} h_{t-1} + b_h$.
$h_t = \tanh(z)$.
I have $\frac{\partial L}{\partial h_t}$. $\frac{\partial L}{\partial z} = \frac{\partial L}{\partial h_t} \times \frac{\partial h_t}{\partial z}$ = dnext_state × $(1 - \tanh^2(z$
$\frac{\partial L}{\partial W_{hx}} = \frac{\partial L}{\partial z} \times \frac{\partial z}{\partial W_{hx}} = dz \times x_t^T$

SOLO

$$z = W_{hx} x_t + W_{hh} h_{t-1} + b_h$$

$$\frac{\partial L}{\partial W_{hx}} = \frac{\partial L}{\partial z} \times \frac{\partial z}{\partial W_{hx}}$$
$$= dz \times (x_t^T)$$

$$\frac{\partial L}{\partial x_t} = \frac{\partial L}{\partial z} \times \frac{\partial z}{\partial x_t}$$
$$= dz \times W_{hx}^T$$

$$\frac{\partial L}{\partial b_h} = dz \times 1.$$

$$\frac{\partial L}{\partial W_{hh}} = \frac{\partial L}{\partial z} \cdot \frac{\partial z}{\partial W_{hh}}$$
$$= dz \times h_{t-1}$$

$$\frac{\partial L}{\partial h_{t-1}} = dz \times W_{hh}$$

Forward Propag RNN:

inp : ~~(batch size, seq length, input dim)~~
init_hidden_state: ~~2D tensor (batch size, hidden dim)~~
input weights .

forward propg rnn:

inp = word_sequence: (2, 3, 4) — capt° ↑ (2), 3 words ↓ (3), 4d. embedding ↑ (4)

initial_context: (2, 5).
⇒ 2 capt°, 5 hidd. dim.
input weights: (4, 5).
map ⇒ 4d emb → 5d hidden states
bias.

From Example

Fwd steps

inp is: [[- - - - - - - - -] [- - - - - - - - -] [- - - - - - - - -]] ⇒ 3x10 matrix
- Row → one embedding.
- Col → size / features of word.

prev_hidden: [[—] [—] [—]] → 3x4 matrix;

word weights: [[—] [—] [—] ⋮ [—]] 10x4 → 4 hidden dim. (vocab size ↑ 10)

memory_weights: [[—] [—] [—] [—]] 4x4

$$z_{hx} = W_{hx} \cdot x + b_h.$$
$$z_{hh} = W_{hh} h_{t-1}$$

$$z = z_{hx} + z_{hh}$$
$$h_t = \tanh(z).$$

Time step here represents = nb. of words in a caption.

SOLO

word_sequence: 1 capt [[- - -] → word 1 [- - -] → word 2 [- - -]] → word 3 — 2 samples, 3 words, 4 emb. dim

1 capt [[- - -] → word 1 [- - -] → word 2 [- - -]]] → word 3

Fwd RNN:

Init. cont: [[- - -] [- - -]] 2×5

⇒ need to iterate through word sequence.

⇒ Update hidden states / weights.

$h = \begin{bmatrix} h_1 \\ h_2 \\ h_5 \end{bmatrix}$: Initial context ≈ initial hidden state

Bwd RNN:

(3 words) $\frac{\partial E_3}{\partial W_{hx}} = \frac{\partial E_3}{\partial Y_3} \times \frac{\partial Y_3}{\partial h_3} \times \frac{\partial h_3}{\partial h_2} \times \frac{\partial h_2}{\partial h_1} \times \frac{\partial h_1}{\partial W_{hx}}$ (dh_3)

Start from end → the beginning. h_0 → fw →

[[[- - -] seqlen×1 [- - -] [- - -]] [[- - -] [- - -] [- - -]]]

W_{hh}, W_{hx}, [i] $H\times 1$, [-] seqlen

[[0.1 0.2 0.3] [0.4 0.5 0.6]] [- - -]]

$h_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

$W_{hx} = \begin{pmatrix} 0.1 & 0. \\ 0.3 & 0. \\ 0.5 & 0. \end{pmatrix}$

$W_{hh} = \begin{pmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{pmatrix}$

add for all "t", all "c".

Forward

$z_t = x_t W_{hxx} + h_{t-1} W_{hh} + b$

$h_t = \tanh(z)$.

$z_0 = (a \;\; b)$; $z_1 = (c \;\; d)$ hidden states $= \begin{pmatrix} x & x \\ x & x \end{pmatrix}_{2\times 2}$

[- - -] → feature vector

Backward:

dhnext = dhstates[t] = (- -).

$dx = dz \times W_{inp}^T$ $dz = 1 - \tanh^2$

$dprevh = dz \times W_{hh}^T$

dw-inp = ...

dw-h = ...

db = ...

Fwd Word Embedding: 5×3:

caption length: 4.

[0 (3) 1 (2)]

⇒ [- - -] [- - -] [- - -] [- - -] 4×3

0, 1, 2, 3, 4

[1 0 0 0]

Bck. Word Embedding:

word_index: x(i,t);

[- - -] [- - -] [- - -]

inp: (2, 3, 10)

init. hidd. state: (2, 5)

W_{hx}: (10, 5)

W_{hh}: (5, 5)

bias: (5, 1).

h: (2, 3, 5).

x_t = 1×10.

next_hidden_state =

[1 2 3 4 5] [1 2 3 4 5] [1 2 3 4 5]

LSTM:

$$\begin{bmatrix} i \\ f \\ o \\ g \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$c_t = f \odot c_{t-1} + i \odot g$

$h_t = o \odot \tanh(c_t)$.

LSTM_step: given: inp, h_{t-1}, c_{t-1}, W_{hx}, W_{hh}, b

At time "t":

$x_t \in \mathbb{R}^D$; $h_{t-1} \in \mathbb{R}^H$; $c_{t-1} \in \mathbb{R}^H$.

need to learn: $W_{hx} \in \mathbb{R}^{4H \times D}$, $W_{hh} \in \mathbb{R}^{4H \times H}$, $b \in \mathbb{R}^{4H}$

(1): $a = W_{hx} x_t + W_{hh} h_{t-1} + b$

(2): split "a" into 4 vecs $\in \mathbb{R}^H$: np.hsplit

a_i = first H elem. a_o: —

a_f = 2nd H elem. a_g: —

$i = \sigma(a_i)$ $o = \sigma(a_o)$

$f = \sigma(a_f)$ $g = \tanh(a_g)$.

!!! DO: $a = x_t W_{hx} + H_{t-1} W_{hh}$.

backward:

sanity: inp: (2, 3, 10) W_{hx} = (10, 5) b_{hh} = (5, 1)

h_0: (2, 5) W_{hh} = (5, 5) h_t = (2, 3, 5)

next h = (2, 5)

t=0:

x_t = 1 x 10.

next h_t, = ((1, 10); (1, 5); (10, 5), (5, 5), (5, 1))

z = 1x10 x 10x5 + (1, 5) x (5, 1) = (1, 5) + (1, 5) = (1, 5)

next_ht = (1, 1)

hidden_states (t, :, :) = next_hidden

backward prop:

x; inp; W_{hx}; W_{hh}; h_0; b

(1, 5); (2, 3, 10); (10, 5); (5, 5); (2, 5); (5, 1).

dinp =

SOLO

$dh_t = (2,5)$; $z\ (2,5)$; $inp = (2,3,10)$; $W_{hx} = (10,5)$; $W_{hh} = (5,5)$; b -

$dz = (2,5)$.

$dinp = dz \times W_{hx}^T \Rightarrow dinp = (2,10)$. $dW_{hh} = (5,5)$.

$dW_{hx} = inp^T dz$ $\quad dW_{hx} = (10,5)$.

$dh_{t-1} = --$ $\quad dh_{t-1} = (10,5)$

backw. rnn:

$dinp = (2,3,10)$; $dW_{hx} = (10,5)$ $\quad dh_t = (2,5)$

$dh_0 = (2,5)$; $dW_{hh} = (5,5)$

$i = 0,1,2$: $\rightarrow dh_{t-1} = [- - - - -]$

$t = 3,2,1$

$dh_{t+1} = dh_t(t) + dh_{t-1}$.

$dx, dh_{t-1}, dW_{hx}\ dW_{hh}, db$ = bckwdstep(dh_{t+1}, cache.

$dx = dh_{t+1}$

$\alpha\ \alpha$

one for loop:

~~t = 3,2,1~~,

t = reverse:

~~dh~~ = ~~dh~~.

$dh(t)$ = dhstates[:, t, :]. $\Rightarrow$ across "t" for all samples.

LSTM: backp. through the

$$\begin{bmatrix} i \\ f \\ o \\ g \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$c_t = f * c_{t-1} + i \odot g$

$h_t = o \odot \tanh(c_t)$.

$$da = \begin{bmatrix} da_i \\ da_f \\ da_o \\ da_g \end{bmatrix} \Big\}\ \text{CODE}$$

Given: dc_t & dh_t.

$dinp = da \times W_{hx}^T$

$dh_{t-1} = da \times W_{hh}^T$

$dc_{t-1} = dc_t * f$

$dW_{hx} = inp^T * da$

$dW_{hh} = h_{t-1}^T * da$

$d$bias

$di = dc_t * g$

$do = dh_t \times \partial h_t/\partial o = dh_t * \tanh(c_t)$

$dg = dc_t \times i$

$dc_t = dc_t + dh_t * o$

$df = dc_t * c_{t-1}$

rnn model:

Input: $Vct_{D\times1}$.

process with Rnn: $h_{H\times1}$

output: W_{ct}; length seqlen.

$h_0 = W \cdot x + bias$

(1) Vct "D" $\rightarrow$ Init. $h_0 \Rightarrow$ (batch, D) $\rightarrow$ (batch, H).

(2) W.E: take indices of captions $\rightarrow$ Feed - WE.

3) seq Processing: