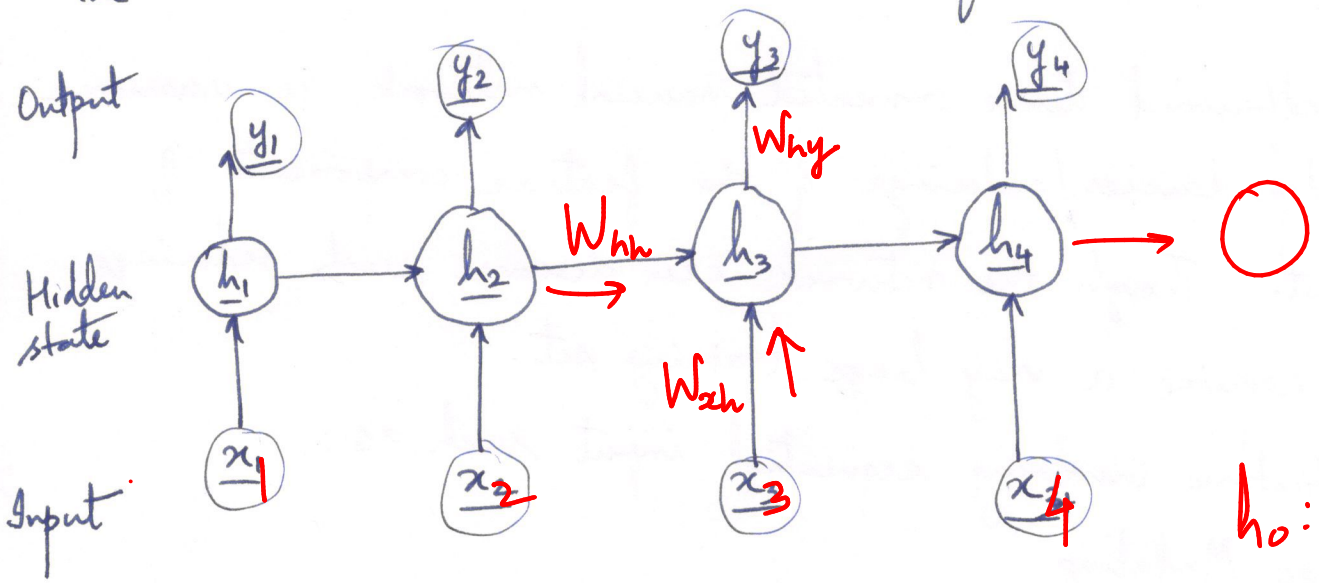
The recurrent network can be unfurled in time



h_0 : initial hidden state

$$h_t = f(h_{t-1}, x_t)$$
$$y_t = g(h_t)$$

$$h_1 = f(h_0, x_1)$$
$$h_2 = f(f(h_0, x_1), x_2)$$
$$\vdots$$

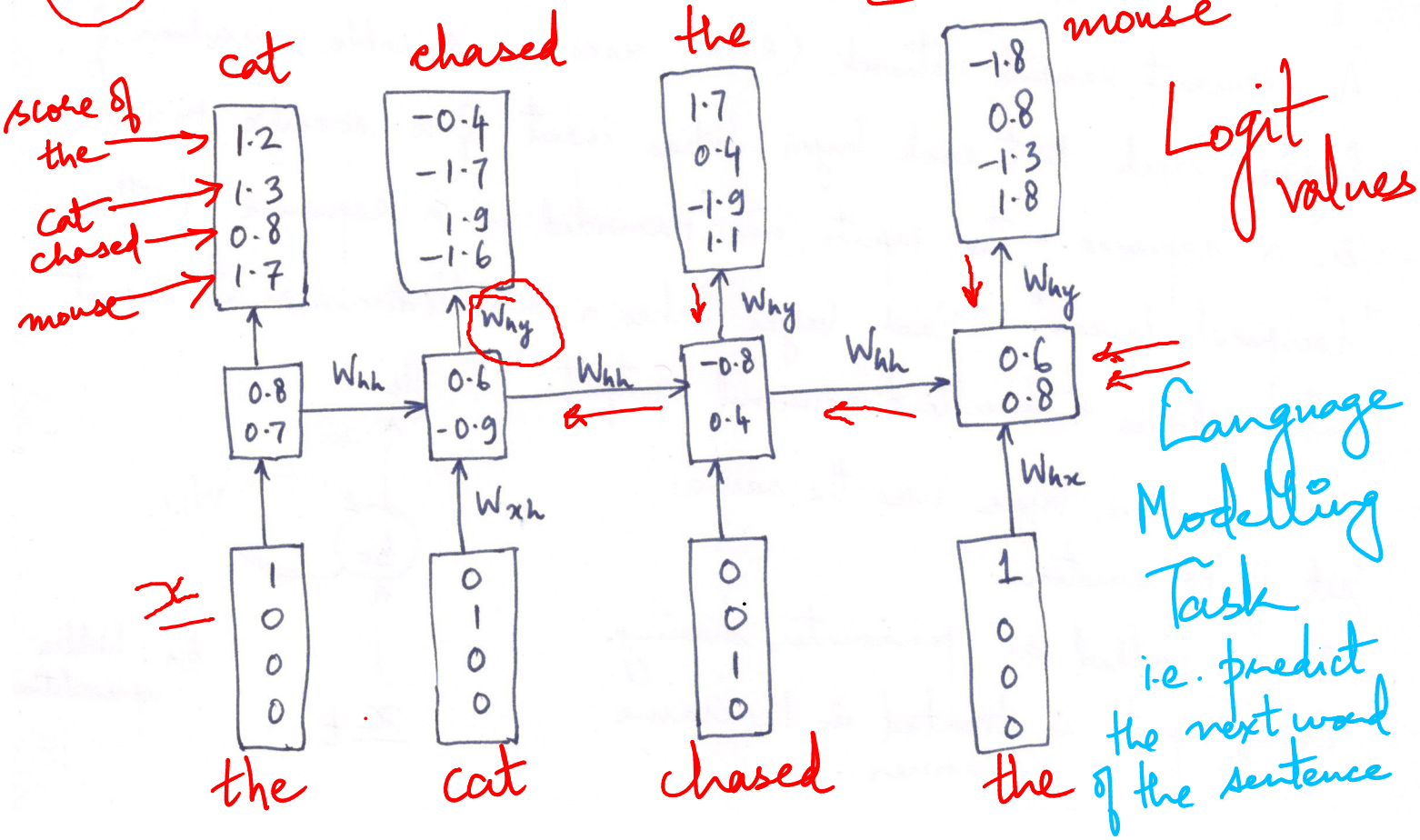
Since $y_t = g(h_t)$

$$y_t = F_t(x_1, x_2, \dots x_t)$$

$$h_t = \tanh(W_{xh} x_t + W_{hh} h_{t-1})$$
$$y_t = W_{hy} h_t$$

tanh: non-linearity

Affine



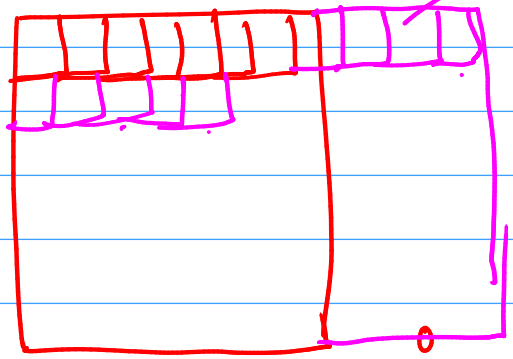
Logit values

Language Modelling Task

i.e. predict the next word of the sentence

2D Grid

CNN

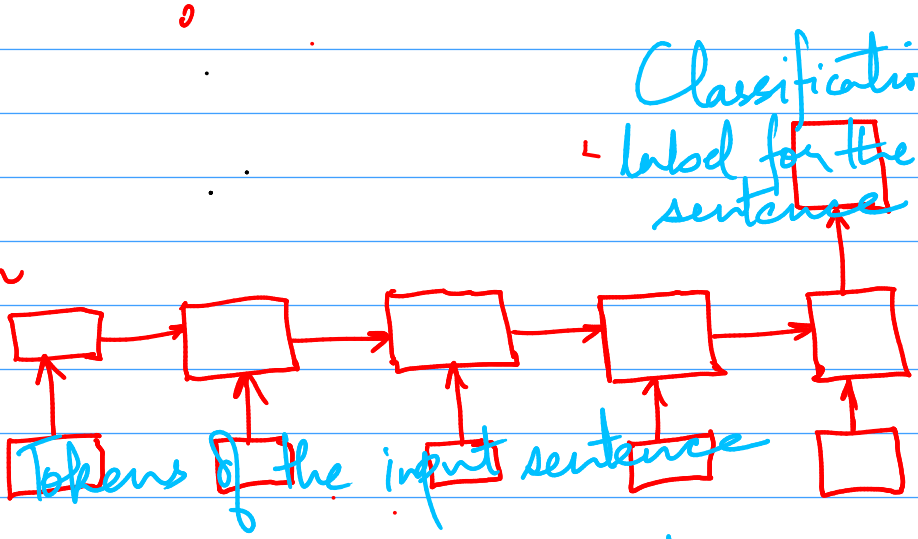


Convolution kernel remains the same

Image Classification

Architectures

Sentence Classification



Machine Translation

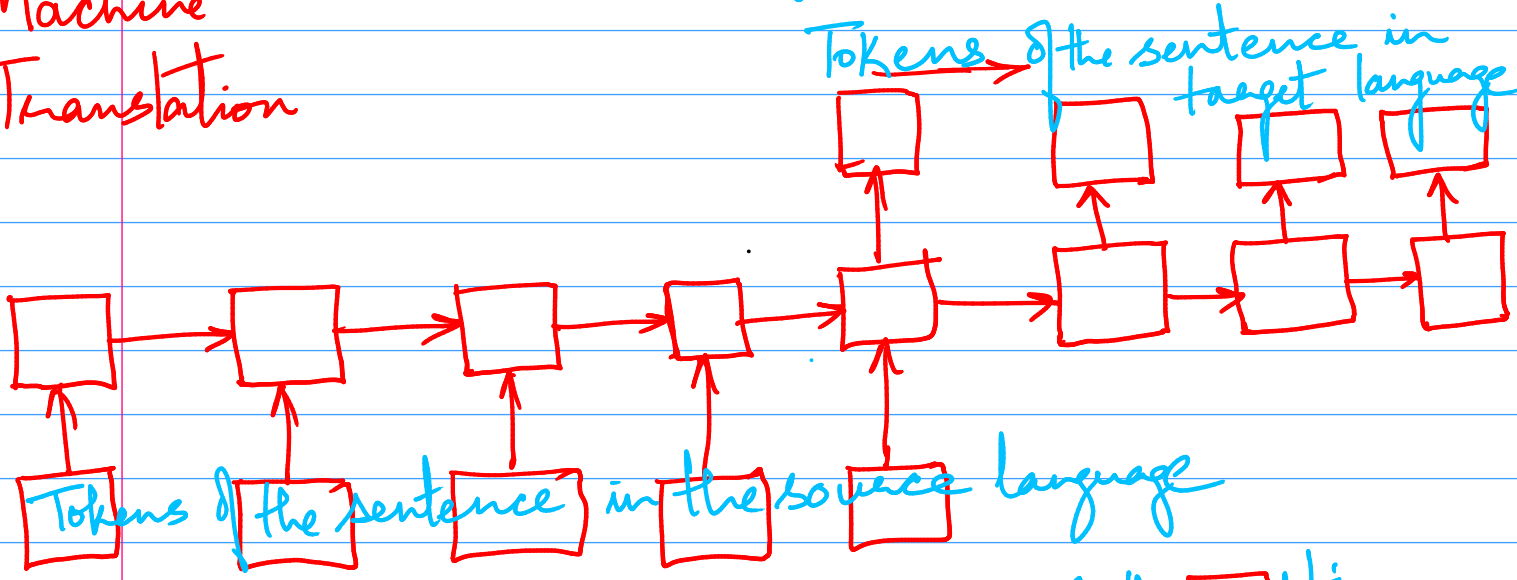
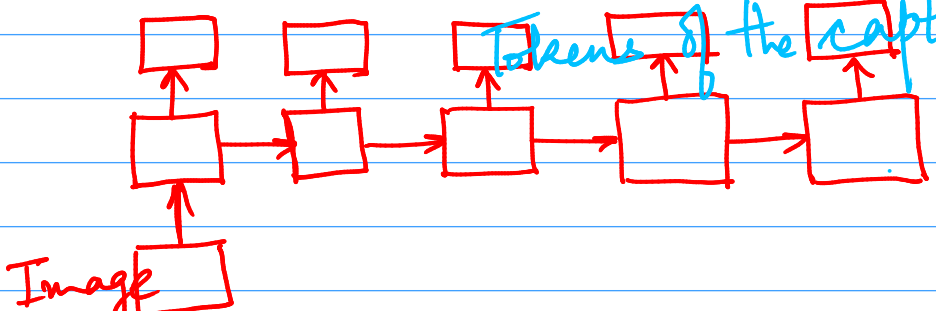


Image Captioning



Backpropagation Through Time

The output vector $y_t \equiv [y_t^1, y_t^2 \ldots y_t^d]$ is converted to a vector of probabilities using a softmax function

$$[p_t^1 \ldots p_t^d] = \text{Softmax}([y_t^1, y_t^2, \ldots y_t^d])$$

The loss function for all T time stamps is

$$L = -\sum_{t=1}^{T} \log(p_t^{j_t})$$

j_t : index of the correct label as given in GT.

The derivative of the loss w.r.t. the raw outputs is computed as

$$\frac{\partial L}{\partial y_t^k} = p_t^k - I(k, j_t)$$

(refer to the derivation of the gradient of the CE loss)

The indicator function I gives an output 1 when k and j_t are the same.

To handle the shared weights we introduce temporal variables $W_{xh}^{(t)}$, $W_{hh}^{(t)}$ and $W_{hy}^{(t)}$ for time-stamp t.

We first perform conventional back propagation pretending that the variables are distinct from one another.

Finally, a unified update for each weight parameter is computed by adding the contributions of the temporal versions of the variables.

Steps

1. Run the input sequentially in the forward direction through time and compute the errors and the negative log loss of softmax layer at each time-stamp.

2. Use conventional back propagation to compute

$$\frac{\partial L}{\partial W_{xh}^{(t)}} \quad \frac{\partial L}{\partial W_{hh}^{(t)}} \quad \frac{\partial L}{\partial W_{hy}^{(t)}}$$

3. Compute

$$\frac{\partial L}{\partial W_{xh}} = \sum_{t=1}^{T} \frac{\partial L}{\partial W_{xh}^{(t)}} \frac{\partial W_{xh}^{(t)}}{\partial W_{xh}}$$

$$\frac{\partial L}{\partial W_{hh}} = \sum_{t=1}^{T} \frac{\partial L}{\partial W_{hh}^{(t)}} \frac{\partial W_{hh}^{(t)}}{\partial W_{hh}}$$

$$\frac{\partial L}{\partial W_{hy}} = \sum_{t=1}^{T} \frac{\partial L}{\partial W_{hy}^{(t)}} \frac{\partial W_{hy}^{(t)}}{\partial W_{hy}}$$

Temporal aggregation wrapped around conventional back propagation

Implicitly we have set $\frac{\partial W_{xh}^{(t)}}{\partial W_{xh}} = 1. = \frac{\partial W_{xh}^{(t)}}{\partial W_{xh}} = \frac{\partial W_{hh}^{(t)}}{\partial W_{hh}} = \frac{\partial W_{hy}^{(t)}}{\partial W_{hy}}$

Truncated Backpropagation through time:

Back propagation updates are done only over segments of the sequence over fixed (modest) length.

Only the portion of the loss over the relevant segment is used to compute the gradients and update the weights.