LSTM Back Propagation with code alongside samples

By Mohit Kumar

LSTM(Long Short Term Memory):Internals:Back Prop

```
#reverse
dh_next = np.zeros_like(h) #dh from the next character
dC_next = np.zeros_like(c)
dx=np.zeros_like(z).T
dy=yhat.copy()
dy=dy-np.reshape(symbols_out_onehot,[-1,1])
dWy=np.dot(dy,h.T)
dBy=dy
dht=np.dot(dy.T,self.WOUT.T).T
for t in reversed(range(seq)):
    z=  zt[t].T
    dht=dht+dh_next
    dot=np.multiply(dht,self.tanh_array(ct[t])*self.dsigmoid(ot[t]))
    dct=np.multiply(dht,ot[t]*self.dtanh(ct[t]))+dC_next
    dcproj=np.multiply(dct,it[t]*(1-cprojt[t]*cprojt[t]))
    dft=np.multiply(dct,coldt[t]*self.dsigmoid(ft[t]))
    dit=np.multiply(dct,cprojt[t]*self.dsigmoid(it[t]))
```

$$\frac{\mathrm{d}E_t}{\mathrm{d}By_t} = \frac{\partial E_t}{\partial \hat{y}_t} . \frac{\partial \hat{y}_t}{\partial pred_t}$$

$$\frac{\mathrm{d}E_t}{\mathrm{d}By_t} = \hat{y}_t - y_t$$

$$E_t = crossEntropyLoss(\hat{y}_t, y_t)$$

$$\hat{y}_t = softmax(pred_t)$$

$$pred_t = (h_t.W_y) + b_y$$

$$h_t = o_t * tanh(c_t)$$

$$c_t = (f_t * c_{t-1}) + (i_t * \widetilde{c}_t)$$

$$o_t = \sigma\left(W_o.[h_{t-1}, x_t] + b_o\right)$$

$$\widetilde{c}_t = \tanh\left(W_c.[h_{t-1}, x_t] + b_c\right)$$

$$i_t = \sigma\left(W_i.[h_{t-1}, x_t] + b_i\right)$$

$$f_t = \sigma\left(W_f.[h_{t-1}, x_t] + b_f\right)$$

LSTM(Long Short Term Memory):Internals:Back Prop

```
#reverse
dh_next = np.zeros_like(h) #dh from the next character
dC_next = np.zeros_like(c)
dx=np.zeros_like(z).T
dy=yhat.copy()
dy=dy-np.reshape(symbols_out_onehot,[-1,1])
dWy=np.dot(dy,h.T)
dBy=dy
dht=np.dot(dy.T,self.WOUT.T).T
for t in reversed(range(seq)):
    z=  zt[t].T
    dht=dht+dh_next
    dot=np.multiply(dht,self.tanh_array(ct[t])*self.dsigmoid(ot[t]))
    dct=np.multiply(dht,ot[t]*self.dtanh(ct[t]))+dC_next
    dcproj=np.multiply(dct,it[t]*(1-cprojt[t]*cprojt[t]))
    dft=np.multiply(dct,coldt[t]*self.dsigmoid(ft[t]))
    dit=np.multiply(dct,cprojt[t]*self.dsigmoid(it[t]))
```

$$\frac{\mathrm{d}E_t}{\mathrm{d}Wy_t} = \frac{\partial E_t}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial pred_t} \cdot \frac{\partial pred_t}{\partial W y_t}$$

$$\frac{\mathrm{d}E_t}{\mathrm{d}Wy_t} = (\hat{y}_t - y_t).h_t$$

$$E_t = crossEntropyLoss(\hat{y}_t, y_t)$$

$$\hat{y}_t = softmax(pred_t)$$

$$pred_t = (h_t.W_y) + b_y$$

$$h_t = o_t * tanh(c_t)$$

$$c_t = (f_t * c_{t-1}) + (i_t * \tilde{c}_t)$$

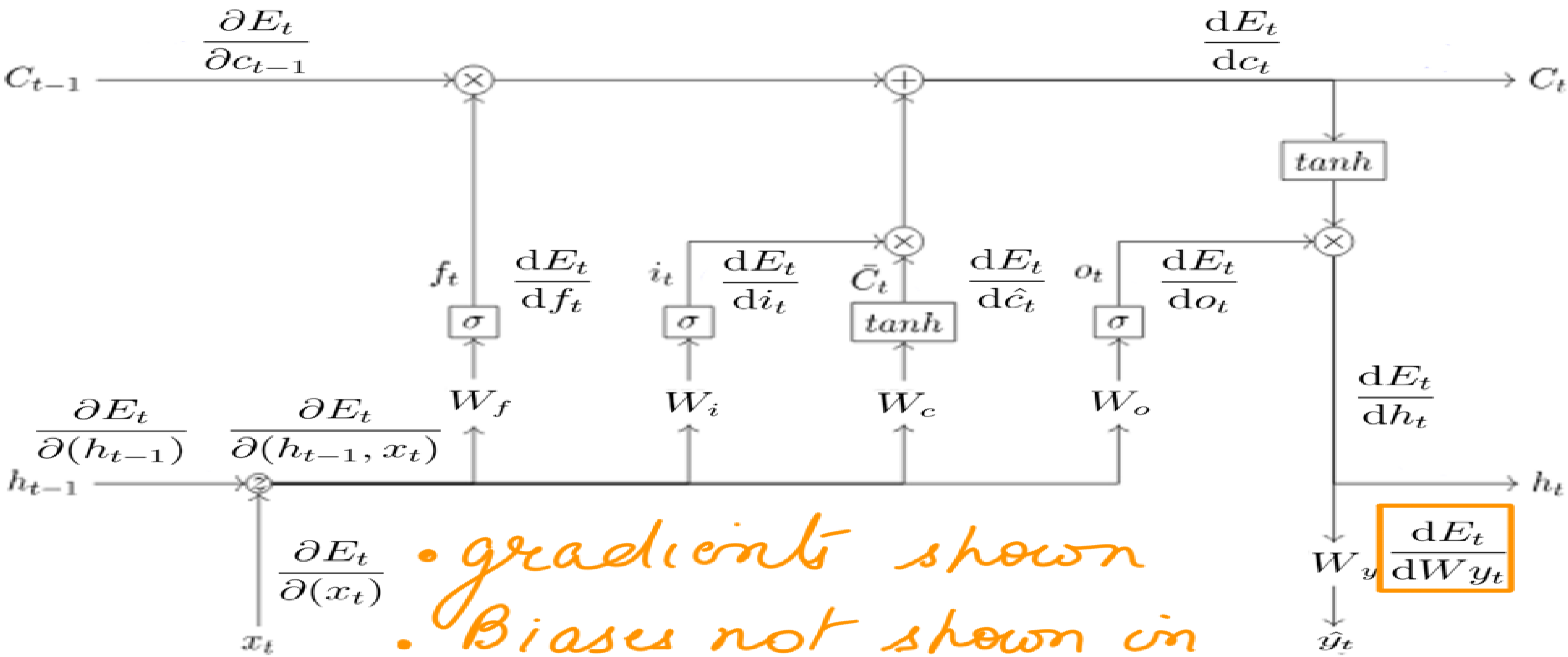
$$o_t = \sigma\left(W_o.[h_{t-1}, x_t] + b_o\right)$$

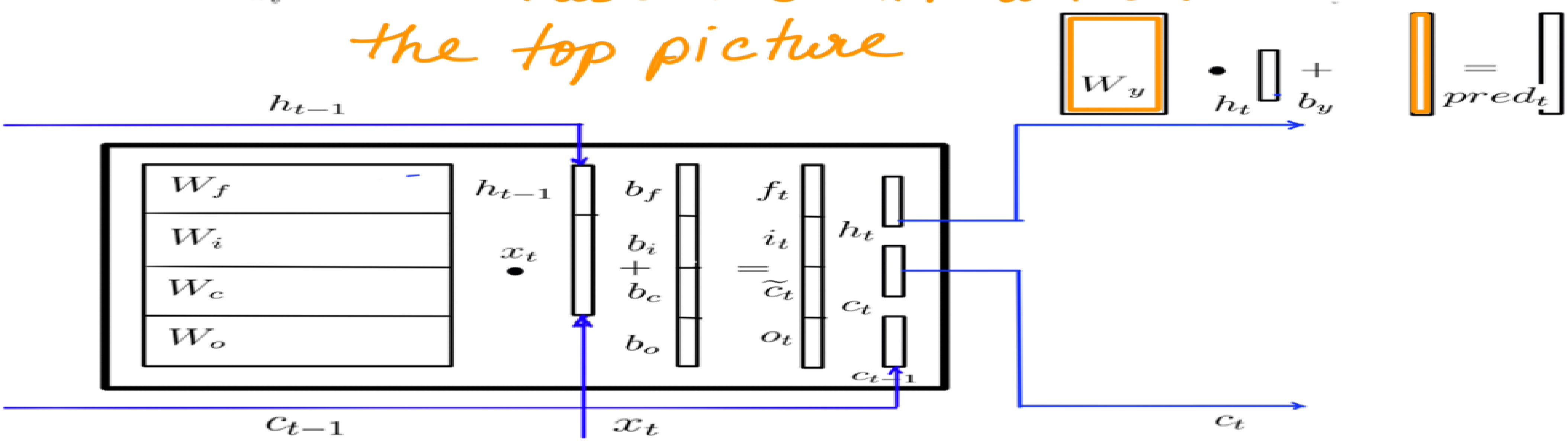
$$\tilde{c}_t = \tanh\left(W_c.[h_{t-1}, x_t] + b_c\right)$$

$$i_t = \sigma\left(W_i.[h_{t-1}, x_t] + b_i\right)$$

$$f_t = \sigma\left(W_f.[h_{t-1}, x_t] + b_f\right)$$

LSTM(Long Short Term Memory):Internals:Back Prop





- gradient shown
- Biases not shown in the top picture

LSTM(Long Short Term Memory):Internals:Back Prop

```
#reverse
dh_next = np.zeros_like(h) #dh from the next character
dC_next = np.zeros_like(c)
dx=np.zeros_like(z).T
dy=yhat.copy()
dy=dy-np.reshape(symbols_out_onehot,[-1,1])
dWy=np.dot(dy,h.T)
dBy=dy
dht=np.dot(dy.T,self.WOUT.T).T
for t in reversed(range(seq)):
    z= zt[t].T
    dht=dht+dh_next
    dot=np.multiply(dht,self.tanh_array(ct[t])*self.dsigmoid(ot[t]))
    dct=np.multiply(dht,ot[t]*self.dtanh(ct[t]))+dC_next
    dcproj=np.multiply(dct,it[t]*(1-cprojt[t]*cprojt[t]))
    dft=np.multiply(dct,coldt[t]*self.dsigmoid(ft[t]))
    dit=np.multiply(dct,cprojt[t]*self.dsigmoid(it[t]))
```

$$\frac{dE_t}{dh_t} = \frac{\partial E_t}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial pred_t} \cdot \frac{\partial pred_t}{\partial h_t}$$

$$\frac{dE_t}{dh_t} = (\hat{y}_t - y_t.W_y) + dh\_next$$

- Wout is Wy.
- Recurrent element zeros to begin with
- Done sequence number of times.

$$E_t = crossEntropyLoss(\hat{y}_t, y_t)$$

$$\hat{y}_t = softmax(pred_t)$$

$$pred_t = (h_t.W_y) + b_y$$

$$h_t = o_t * tanh(c_t)$$

$$c_t = (f_t * c_{t-1}) + (i_t * \tilde{c}_t)$$

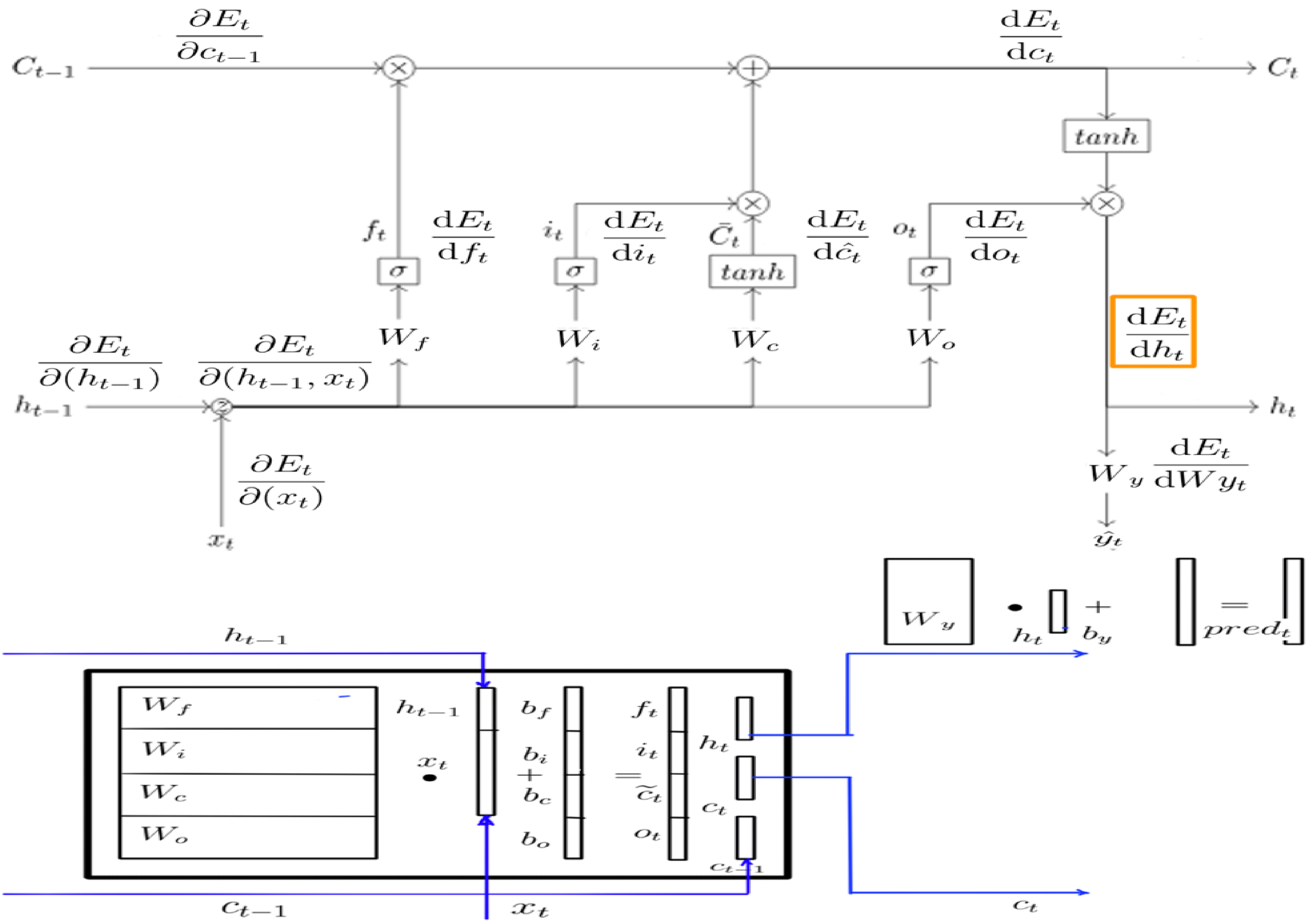
$$o_t = \sigma\left(W_o.[h_{t-1}, x_t] + b_o\right)$$

$$\tilde{c}_t = \tanh\left(W_c.[h_{t-1}, x_t] + b_c\right)$$

$$i_t = \sigma\left(W_i.[h_{t-1}, x_t] + b_i\right)$$

$$f_t = \sigma\left(W_f.[h_{t-1}, x_t] + b_f\right)$$

LSTM(Long Short Term Memory):Internals:Back Prop



LSTM(Long Short Term Memory):Internals:Back Prop

```
#reverse
dh_next = np.zeros_like(h) #dh from the next characte
dC_next = np.zeros_like(c)
dx=np.zeros_like(z).T
dy=yhat.copy()
dy=dy-np.reshape(symbols_out_onehot,[-1,1])
dWy=np.dot(dy,h.T)
dBy=dy
dht=np.dot(dy.T,self.WOUT.T).T
for t in reversed(range(seq)):
    z=  zt[t].T
    dht=dht+dh_next
    dot=np.multiply(dht,self.tanh_array(ct[t])*self.dsigmoid(ot[t]))
    dct=np.multiply(dht,ot[t]*self.dtanh(ct[t]))+dC_next
    dcproj=np.multiply(dct,it[t]*(1-cprojt[t]*cprojt[t]))
    dft=np.multiply(dct,coldt[t]*self.dsigmoid(ft[t]))
    dit=np.multiply(dct,cprojt[t]*self.dsigmoid(it[t]))
```

$$\frac{\mathrm{d}E_t}{\mathrm{d}o_t} = \frac{\partial E_t}{\partial h_t} . \frac{\partial h_t}{\partial o_t}$$

$$\frac{\mathrm{d}E_t}{\mathrm{d}o_t} = \frac{\partial E_t}{\partial h_t} .tanh(c_t).dsigmoid(o_t)$$

```
def dsigmoid(self,f):
    return f*(1-f)
```

$$E_t = crossEntropyLoss(\hat{y}_t, y_t)$$

$$\hat{y}_t = softmax(pred_t)$$

$$pred_t = (h_t.W_y) + b_y$$

$$h_t = o_t * tanh(c_t)$$

$$c_t = (f_t * c_{t-1}) + (i_t * \tilde{c}_t)$$

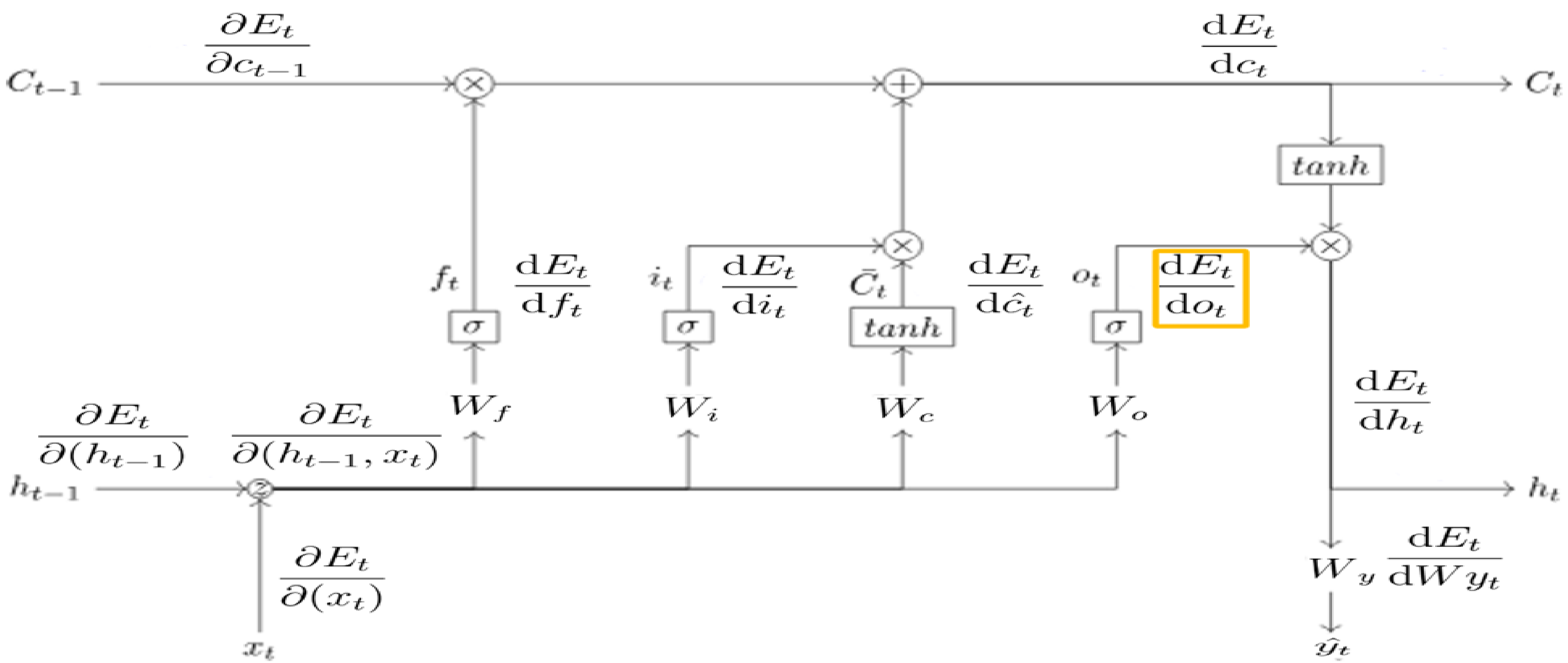
$$o_t = \sigma\left(W_o.[h_{t-1}, x_t] + b_o\right)$$

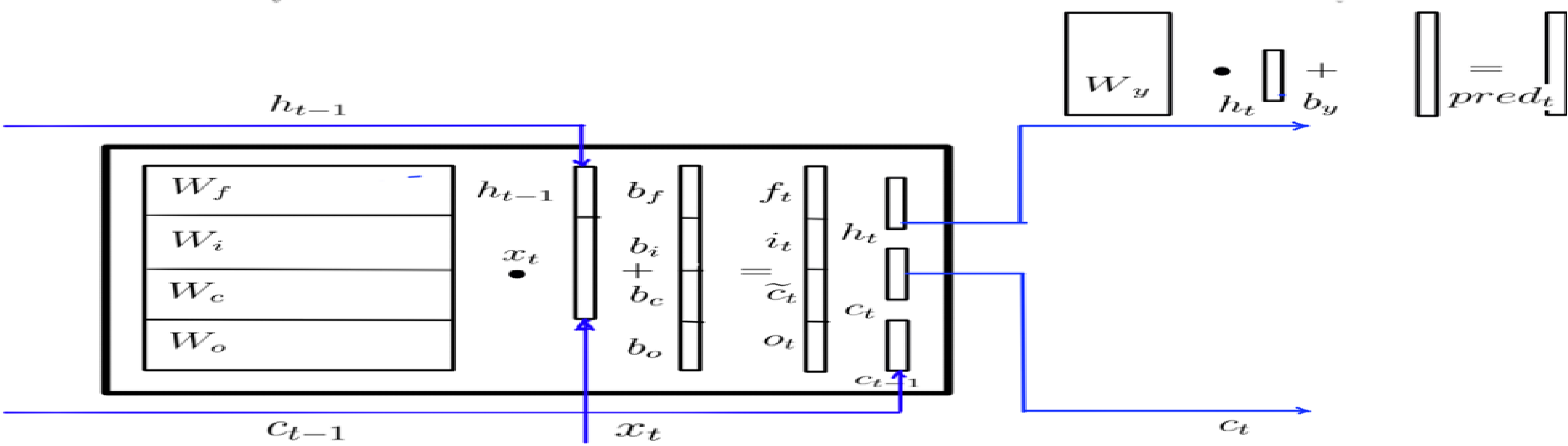
$$\tilde{c}_t = \tanh\left(W_c.[h_{t-1}, x_t] + b_c\right)$$

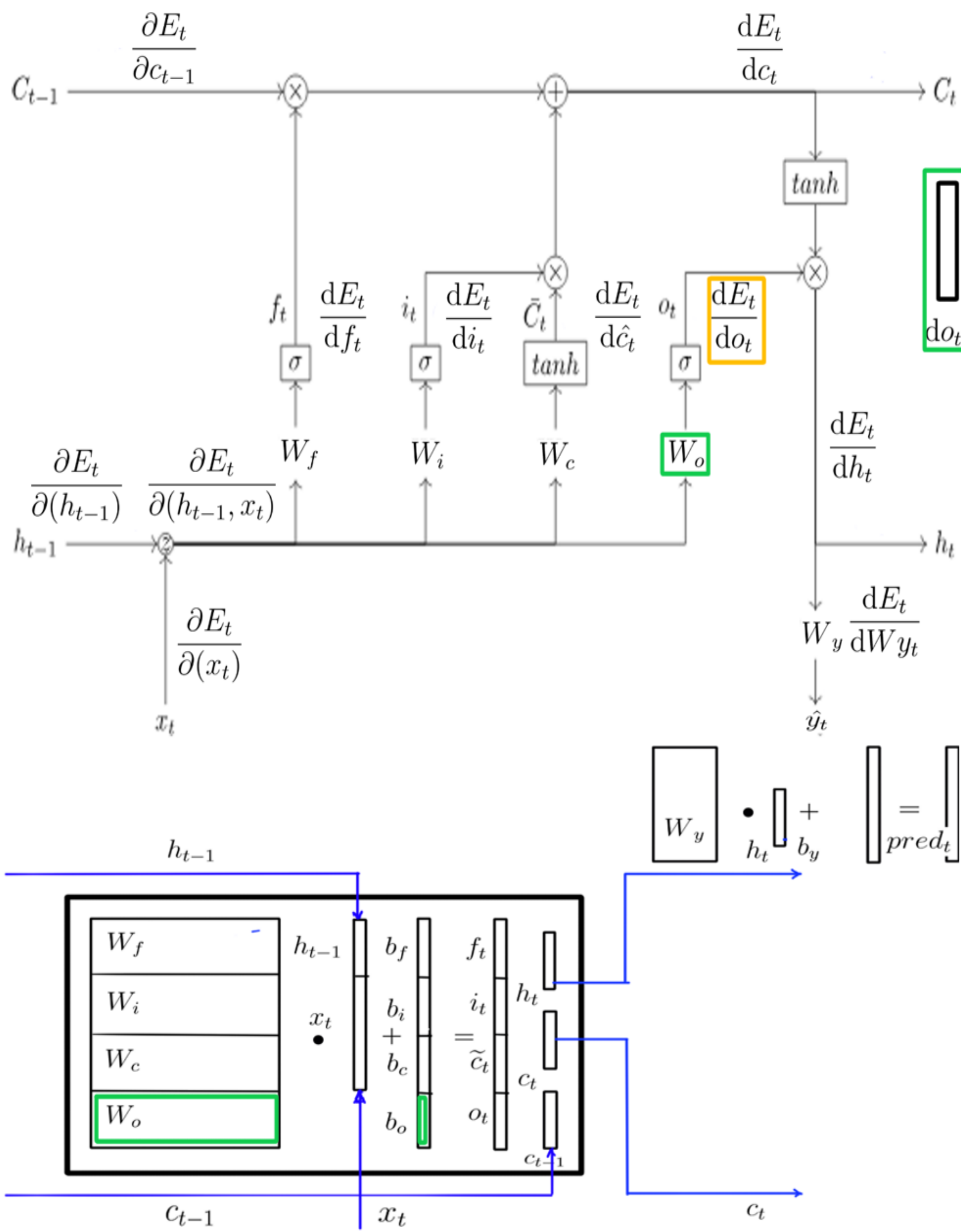
$$i_t = \sigma\left(W_i.[h_{t-1}, x_t] + b_i\right)$$

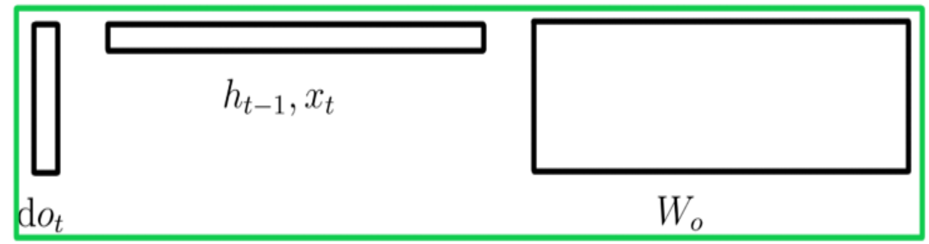
$$f_t = \sigma\left(W_f.[h_{t-1}, x_t] + b_f\right)$$

LSTM(Long Short Term Memory):Internals:Back Prop










```
dwo+=np.dot(dot,z)
dbo+=dot
dxo=np.dot(dot.T,self.WLSTM[self.hidden_size*3:self.hidden_size*4,1:])
```

LSTM(Long Short Term Memory):Internals:Back Prop

```python
#reverse
dh_next = np.zeros_like(h) #dh from the next char
dC_next = np.zeros_like(c)
dx=np.zeros_like(z).T
dy=yhat.copy()
dy=dy-np.reshape(symbols_out_onehot,[-1,1])
dWy=np.dot(dy,h.T)
dBy=dy
dht=np.dot(dy.T,self.WOUT.T).T
for t in reversed(range(seq)):
    z= zt[t].T
    dht=dht+dh_next
    dot=np.multiply(dht,self.tanh_array(ct[t])*self.dsigmoid(ot[t]))
    dct=np.multiply(dht,ot[t]*self.dtanh(ct[t]))+dC_next
    dcproj=np.multiply(dct,it[t]*(1-cprojt[t]*cprojt[t]))
    dft=np.multiply(dct,coldt[t]*self.dsigmoid(ft[t]))
    dit=np.multiply(dct,cprojt[t]*self.dsigmoid(it[t]))
```

$$\frac{dE_t}{dc_t} = \frac{\partial E_t}{\partial h_t} \cdot \frac{\partial h_t}{\partial c_t}$$

$$\frac{dE_t}{dc_t} = \frac{\partial E_t}{\partial h_t} . o_t . dtanh(c_t) + dc\_next$$

- Because of recurrent cell state $c_{(t-1)}$.

```python
def dtanh(self,f):
    tanhf=np.tanh(f)
    return 1 - tanhf * tanhf
```

$$E_t = crossEntropyLoss(\hat{y}_t, y_t)$$

$$\hat{y}_t = softmax(pred_t)$$

$$pred_t = (h_t.W_y) + b_y$$

$$h_t = o_t * tanh(c_t)$$

$$c_t = (f_t * c_{t-1}) + (i_t * \tilde{c}_t)$$

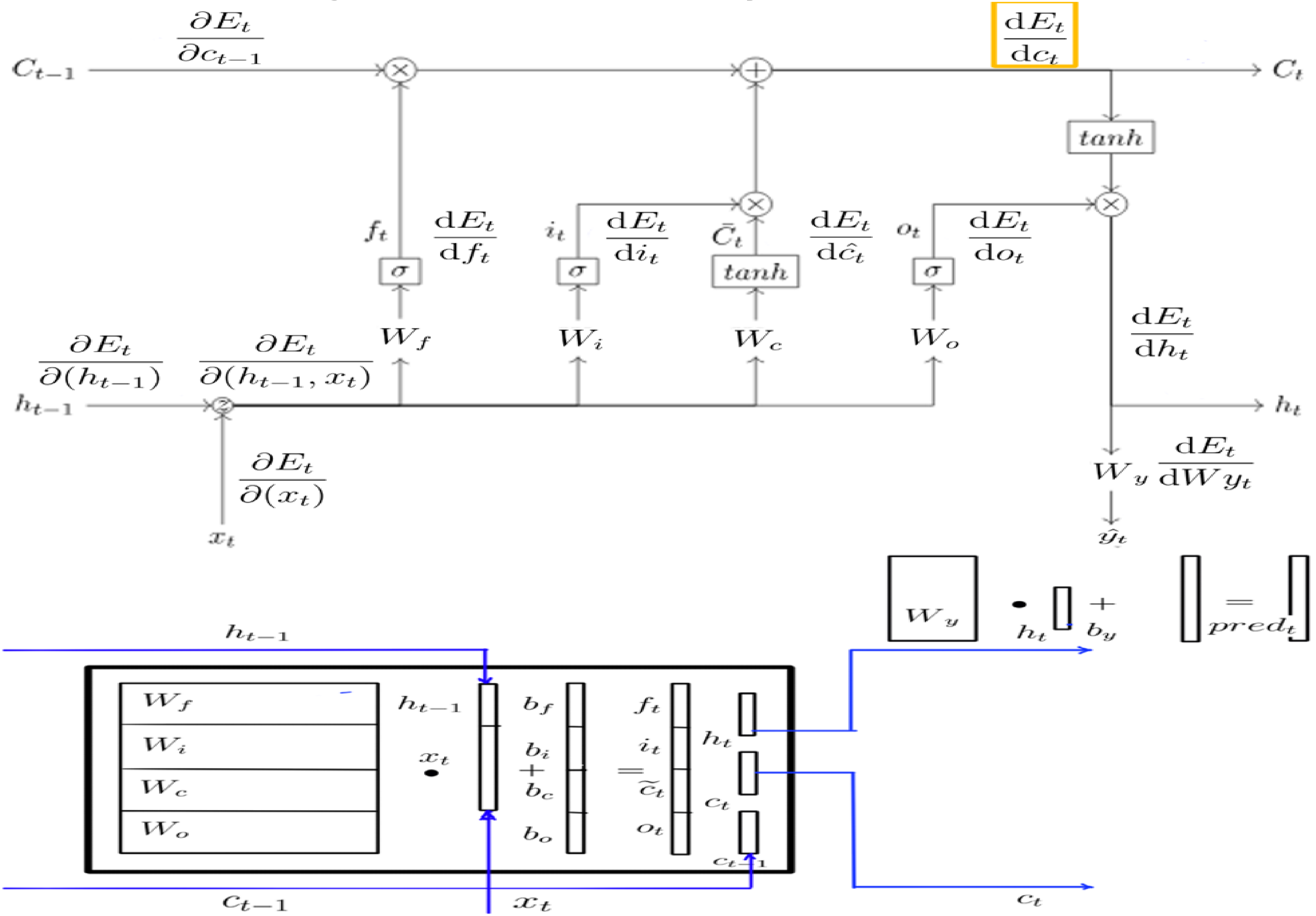
$$o_t = \sigma\left(W_o.[h_{t-1}, x_t] + b_o\right)$$

$$\tilde{c}_t = \tanh\left(W_c.[h_{t-1}, x_t] + b_c\right)$$

$$i_t = \sigma\left(W_i.[h_{t-1}, x_t] + b_i\right)$$

$$f_t = \sigma\left(W_f.[h_{t-1}, x_t] + b_f\right)$$

LSTM(Long Short Term Memory):Internals:Back Prop



LSTM(Long Short Term Memory):Internals:Back Prop

```
#reverse
dh_next = np.zeros_like(h) #dh from the next character
dC_next = np.zeros_like(c)
dx=np.zeros_like(z).T
dy=yhat.copy()
dy=dy-np.reshape(symbols_out_onehot,[-1,1])
dWy=np.dot(dy,h.T)
dBy=dy
dht=np.dot(dy.T,self.WOUT.T).T
for t in reversed(range(seq)):
    z= zt[t].T
    dht=dht+dh_next
    dot=np.multiply(dht,self.tanh_array(ct[t])*self.dsigmoid(ot[t]))
    dct=np.multiply(dht,ot[t]*self.dtanh(ct[t]))+dC_next
    dcproj=np.multiply(dct,it[t]*(1-cprojt[t]*cprojt[t]))
    dft=np.multiply(dct,coldt[t]*self.dsigmoid(ft[t]))
    dit=np.multiply(dct,cprojt[t]*self.dsigmoid(it[t]))
```

$$\frac{\mathrm{d}E_t}{\mathrm{d}\hat{c}_t} = \frac{\partial E_t}{\partial c_t} . \frac{\partial c_t}{\partial \hat{c}_t}$$

$$\frac{\mathrm{d}E_t}{\mathrm{d}\hat{c}_t} = \frac{\partial E_t}{\partial c_t} . i_t . dtanh(\hat{c}_t)$$

- Previously calculated dct
- inlined dtanh
- $\hat{c}$ is cproj.

$$E_t = crossEntropyLoss(\hat{y}_t, y_t)$$

$$\hat{y}_t = softmax(pred_t)$$

$$pred_t = (h_t.W_y) + b_y$$

$$h_t = o_t * tanh(c_t)$$

$$c_t = (f_t * c_{t-1}) + (i_t * \widetilde{c}_t)$$

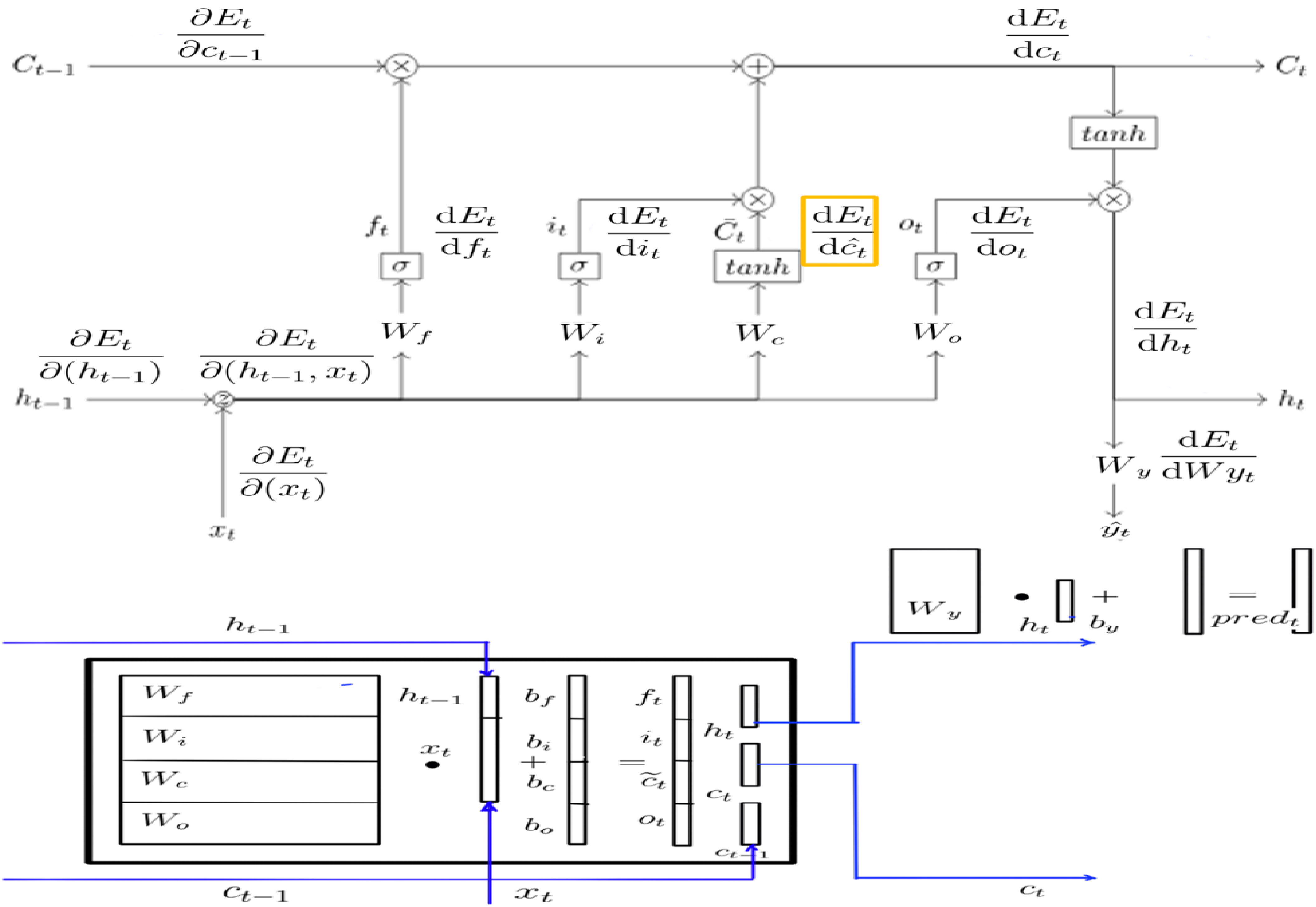
$$o_t = \sigma\left(W_o.[h_{t-1}, x_t] + b_o\right)$$

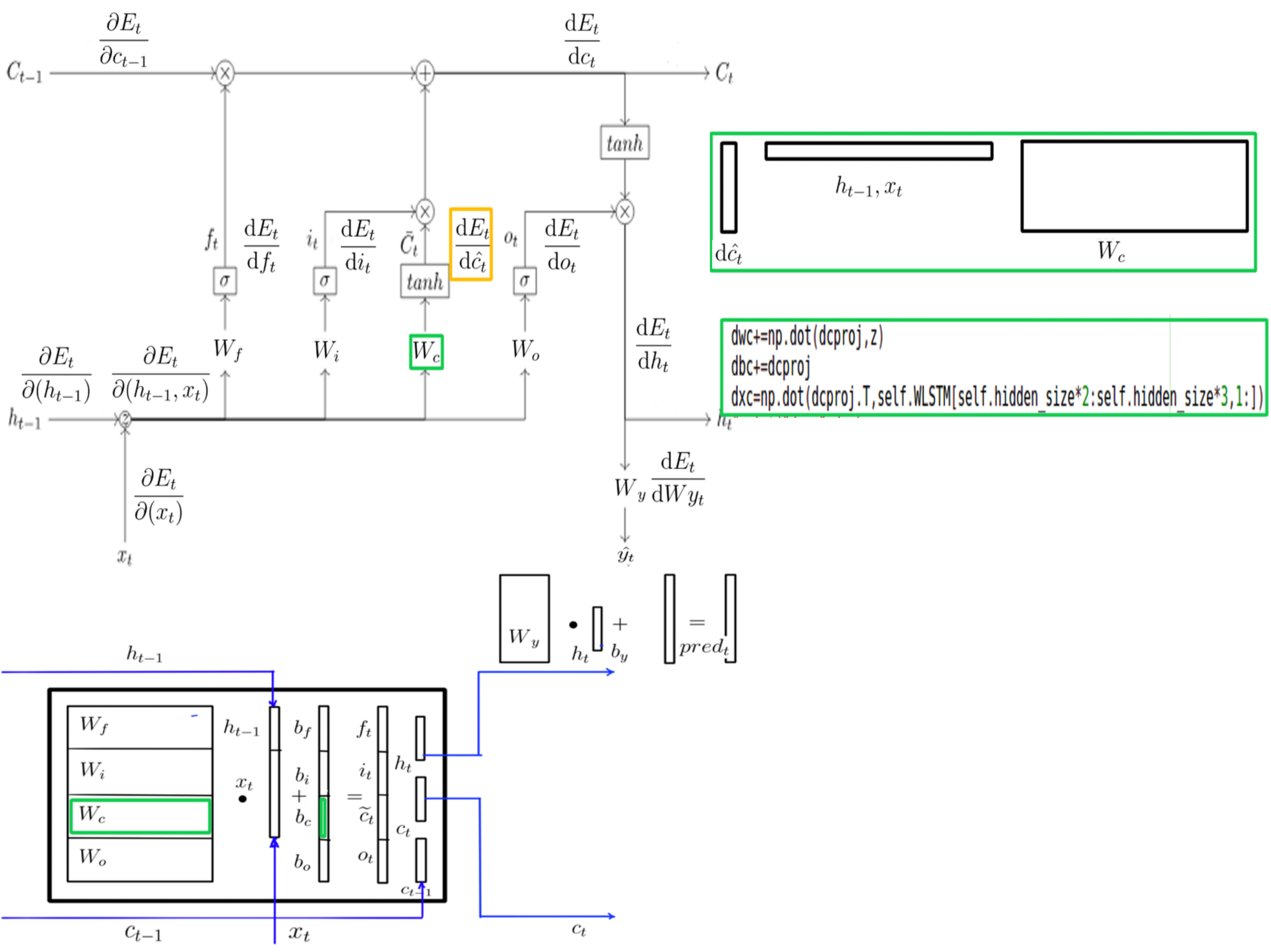
$$\widetilde{c}_t = \tanh\left(W_c.[h_{t-1}, x_t] + b_c\right)$$

$$i_t = \sigma\left(W_i.[h_{t-1}, x_t] + b_i\right)$$

$$f_t = \sigma\left(W_f.[h_{t-1}, x_t] + b_f\right)$$

LSTM(Long Short Term Memory):Internals:Back Prop




$\frac{\partial E_t}{\partial c_{t-1}}$
$\frac{dE_t}{dc_t}$
C_{t-1}
C_t
tanh
h_{t-1}, x_t
$d\hat{c}_t$
W_c
f_t
$\frac{dE_t}{df_t}$
i_t
$\frac{dE_t}{di_t}$
$\bar{C}_t$
$\frac{dE_t}{d\hat{c}_t}$
o_t
$\frac{dE_t}{do_t}$
σ
tanh
$\frac{dE_t}{dh_t}$
dwc+=np.dot(dcproj,z)
dbc+=dcproj
dxc=np.dot(dcproj.T,self.WLSTM[self.hidden_size*2:self.hidden_size*3,1:])
$\frac{\partial E_t}{\partial(h_{t-1})}$
$\frac{\partial E_t}{\partial(h_{t-1}, x_t)}$
W_f
W_i
W_c
W_o
h_{t-1}
h_t
$\frac{\partial E_t}{\partial(x_t)}$
W_y
$\frac{dE_t}{dWy_t}$
x_t
$\hat{y}_t$
W_y
h_t
b_y
$pred_t$
h_{t-1}
W_f
W_i
W_c
W_o
h_{t-1}
x_t
b_f
b_i
b_c
b_o
f_t
i_t
$\tilde{c}_t$
o_t
h_t
c_t
c_{t-1}
c_{t-1}
x_t
c_t