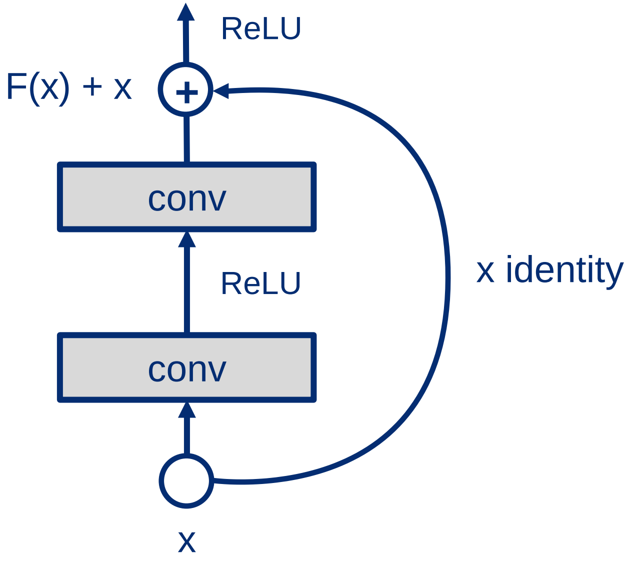

ME
RESNET 50
FASHION MNIST
flip.com

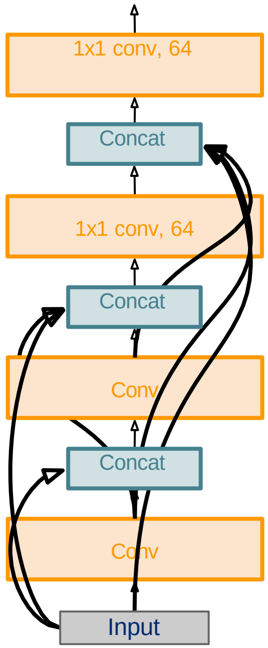

DUCK COMICS
AW YISS
SOME MOTHA FUCKIN polygons
NICE DUCKIE


When you don't use LSTM
for a long sequence RNN:
IN TERMS OF GRADIENTS,
WE HAVE NO GRADIENTS.
imgflip.com

Gradient Flow



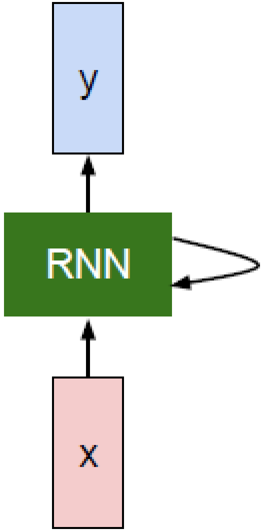
Residual block



Dense block

RNNs

Recurrent Neural Networks: Standard Architecture



$$h_t = f_W(h_{t-1}, x_t)$$

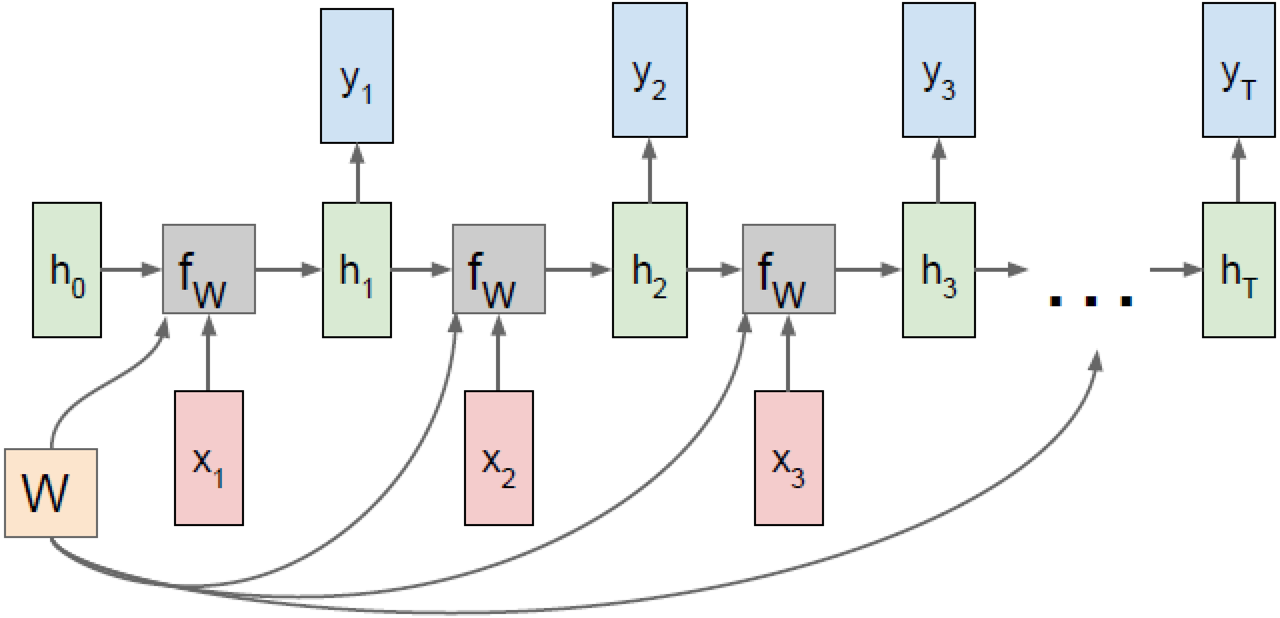
$$\downarrow$$

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

$$y_t = W_{hy}h_t$$

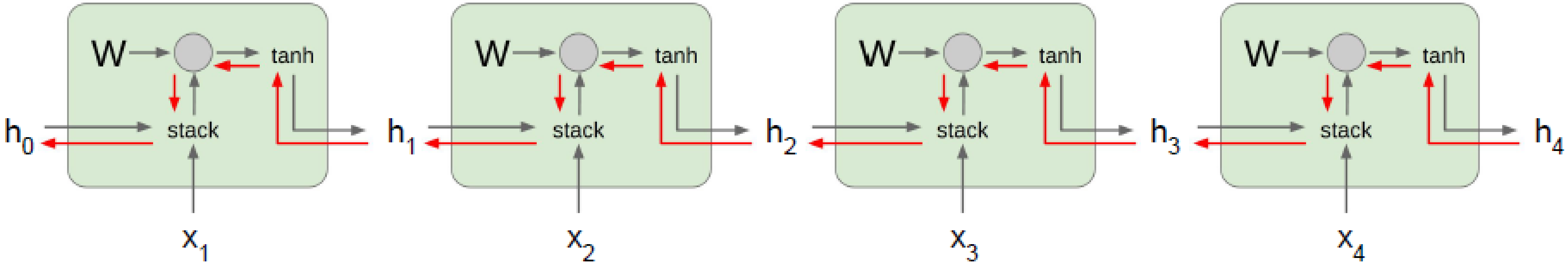
RNNs

Recurrent Neural Networks: Unrolling → Same weights at every timestep!



Gradients

Recurrent Neural Networks: Gradient flow



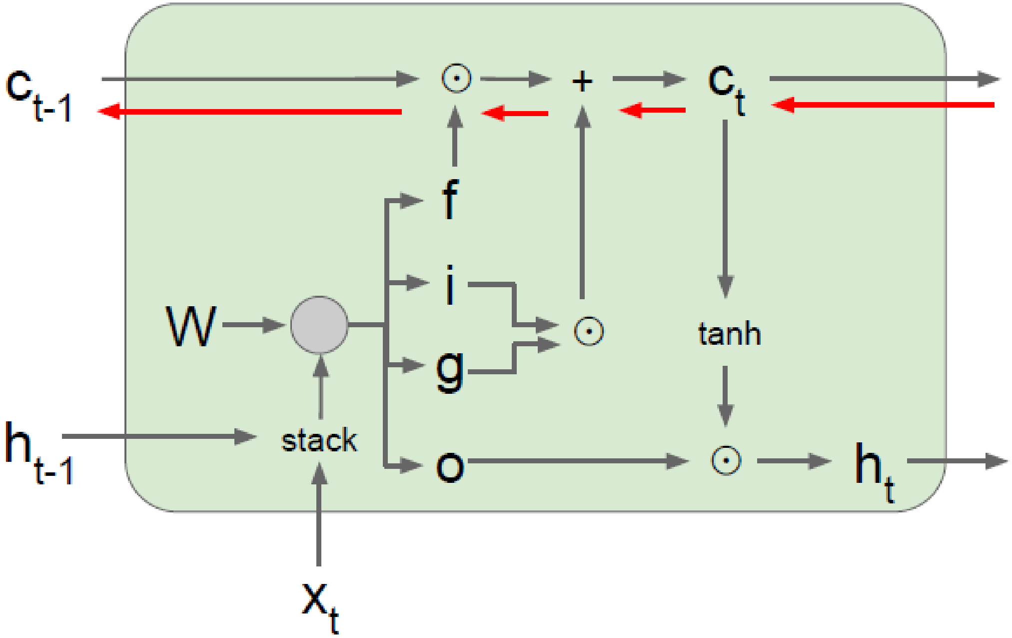
Computing gradient of h_0 involves many factors of W (and repeated tanh)

Largest singular value > 1:
Exploding gradients

Largest singular value < 1:
Vanishing gradients

LSTM

Long Short Term Memory: Mitigating the gradient problem



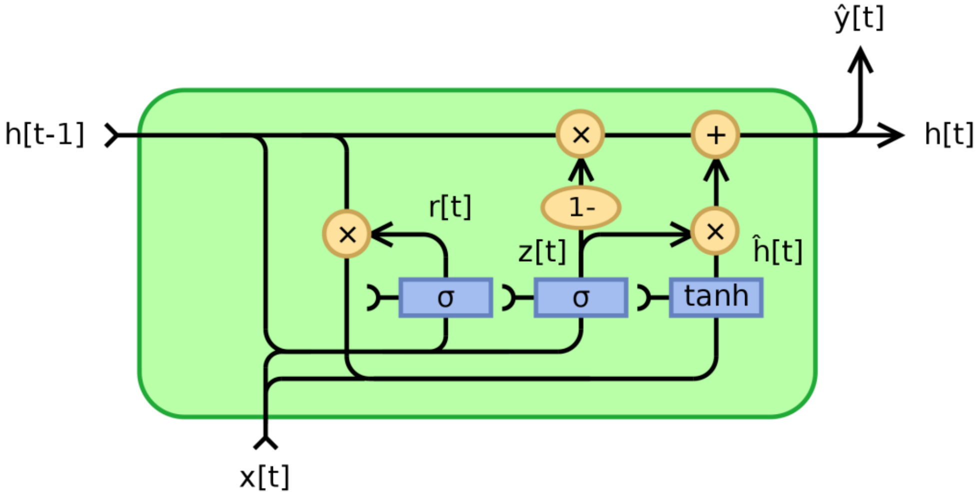
$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

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

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

Gated Recurrent Units

Similar to LSTM without output gate



Variables

- x_t: input vector
- h_t: output vector
- $\hat{h}_t$: candidate activation vector
- z_t: update gate vector
- r_t: reset gate vector
- W, U and b: parameter matrices and vector

Activation functions

- σ_g: The original is a sigmoid function.
- ϕ_h: The original is a hyperbolic tangent.

$$
\begin{aligned}
z_t &= \sigma_g(W_z x_t + U_z h_{t-1} + b_z) \\
r_t &= \sigma_g(W_r x_t + U_r h_{t-1} + b_r) \\
\hat{h}_t &= \phi_h(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \\
h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t
\end{aligned}
$$

Taken from here.