# Learning Long-Term Dependencies with RNN

Roger Gilbertson (CC)

Some slides from Richard Socher, Geoffrey Hinton, Andrej Karpathy

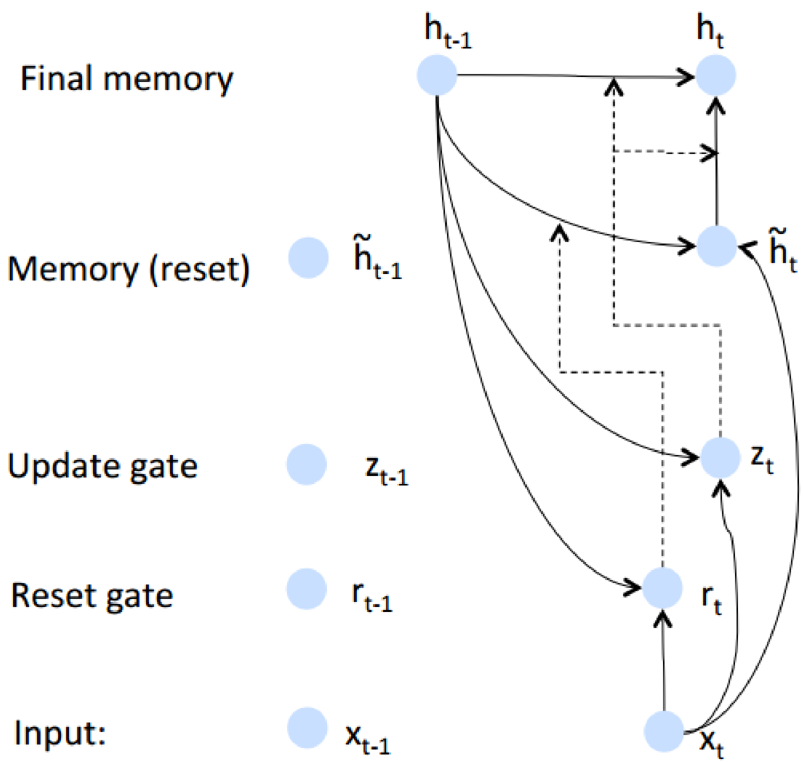
CSC321: Intro to Machine Learning and Neural Networks, Winter 2016

Michael Guerzhoy

# Gated Recurrent Units (GRU)

- Instead of $h_t = tanh(W^{(hh)}h_{t-1} + W^{(hx)}x_t)$ do
  - Update gate: $z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1})$
  - Reset gate: $r_t = \sigma\left(W^{(r)}x_t + U^{(r)}h_{t-1}\right)$
  - New memory: $\tilde{h}_t = \tanh\left(W^{(hx)}x_t + r \circ W^{(hh)}h_{t-1}\right)$
  - Final memory: $h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$
- If update gate is around 0, previous memory is ignored, and only new information is stored
- The reset gate controls whether the input or the previous state determines the current state

# GRU



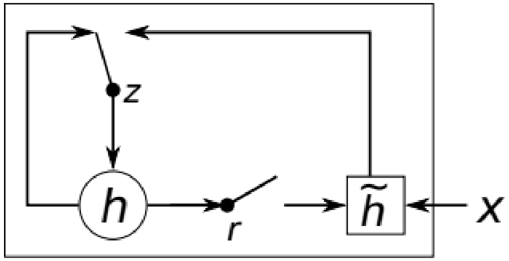

$$z_t = \sigma\left(W^{(z)}x_t + U^{(z)}h_{t-1}\right)$$

$$r_t = \sigma\left(W^{(r)}x_t + U^{(r)}h_{t-1}\right)$$

$$\tilde{h}_t = \tanh\left(Wx_t + r_t \circ Uh_{t-1}\right)$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$$

# GRU intuition




- If reset is close to 0, ignore previous hidden state
  - Allows model to drop information that is irrelevant in the future

$$z_t = \sigma\left(W^{(z)}x_t + U^{(z)}h_{t-1}\right)$$
$$r_t = \sigma\left(W^{(r)}x_t + U^{(r)}h_{t-1}\right)$$
$$\tilde{h}_t = \tanh\left(Wx_t + r_t \circ Uh_{t-1}\right)$$
$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$$

- Update gate z controls how much the past state should matter now
- Units with short-term dependencies will have active reset gates r
- Units with long term dependencies have active update gates z

# Why do GRUs help with the vanishing gradient problem?

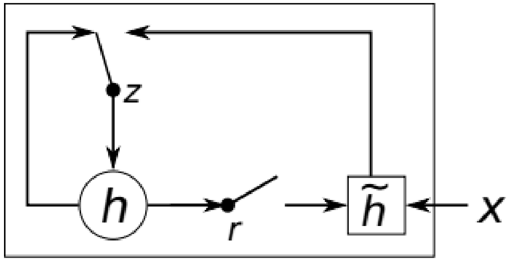
- *We had:*
  - $\frac{\partial J^{(t)}}{\partial W} = \sum_{k=1}^{t} \frac{\partial J^{(t)}}{\partial y_t} \frac{\partial y_t}{\partial W} = \sum_{k=1}^{t} \frac{\partial J^{(t)}}{\partial \hat{y}_k} \frac{\hat{y}_k}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W}$
  - $\frac{\partial h_t}{\partial h_k} = \prod_{j=k+1}^{t} \frac{\partial h_j}{\partial h_{j-1}}$ $\leq \alpha^{t-j-1}$
- *Now:*
  - $\frac{\partial h_j}{\partial h_{j-1}} = z_j + \left(1 - z_j\right) \frac{\partial \widetilde{h}_j}{\partial h_{j-1}}$
  - $\frac{\partial h_j}{\partial h_{j-1}}$ is 1 for $z_j = 1$

$$z_t = \sigma\left(W^{(z)} x_t + U^{(z)} h_{t-1}\right)$$
$$r_t = \sigma\left(W^{(r)} x_t + U^{(r)} h_{t-1}\right)$$
$$\tilde{h}_t = \tanh\left(W x_t + r_t \circ U h_{t-1}\right)$$
$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$$

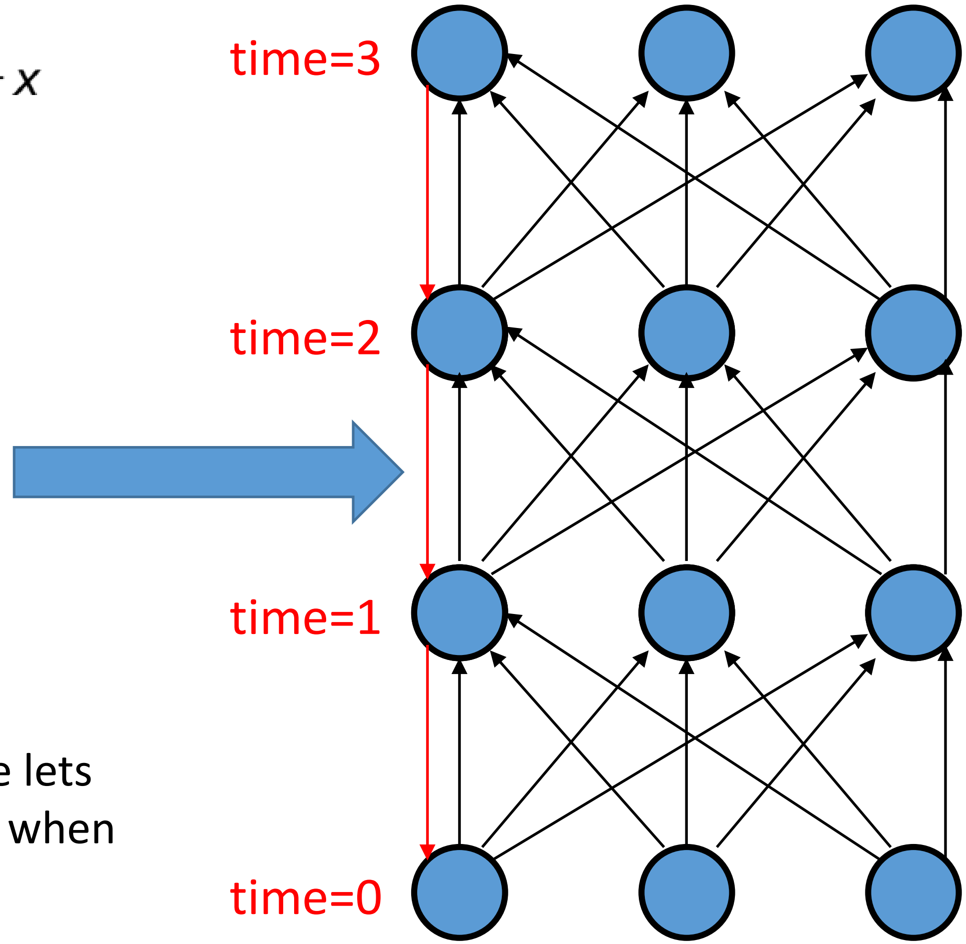
$$z_t = \sigma\left(W^{(z)}x_t + U^{(z)}h_{t-1}\right)$$
$$r_t = \sigma\left(W^{(r)}x_t + U^{(r)}h_{t-1}\right)$$
$$\tilde{h}_t = \tanh\left(Wx_t + r_t \circ Uh_{t-1}\right)$$
$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$$

- $\frac{\partial \tilde{h}_j}{\partial h_{j-1}} = \frac{\partial}{\partial h_{j-1}} \tanh\left(Wx_j + r_j \circ Uh_{j-1}\right) = (1 - \tilde{h}_j^2)(r_j \circ U)$

- $\frac{\partial h_j}{\partial h_{j-1}} = z_j + \left(1 - z_j\right)\frac{\partial \tilde{h}_j}{\partial h_{j-1}}$ is 1 for $z_j = 1$
- $\frac{\partial h_j}{\partial h_{j-1}} = z_j + \left(1 - z_j\right)\frac{\partial \tilde{h}_j}{\partial h_{j-1}}$ is $z_j$ for $r_j = 0$




$z_j = 1$ ➡ ignore




“Shutting” the update gate lets us essentially “skip” layers when calculating the gradient.

This ameliorates the vanishing, exploding gradient problem.

$$\frac{\partial h_t}{\partial h_k} = \prod_{j=k+1}^{t} \frac{\partial h_j}{\partial h_{j-1}}$$

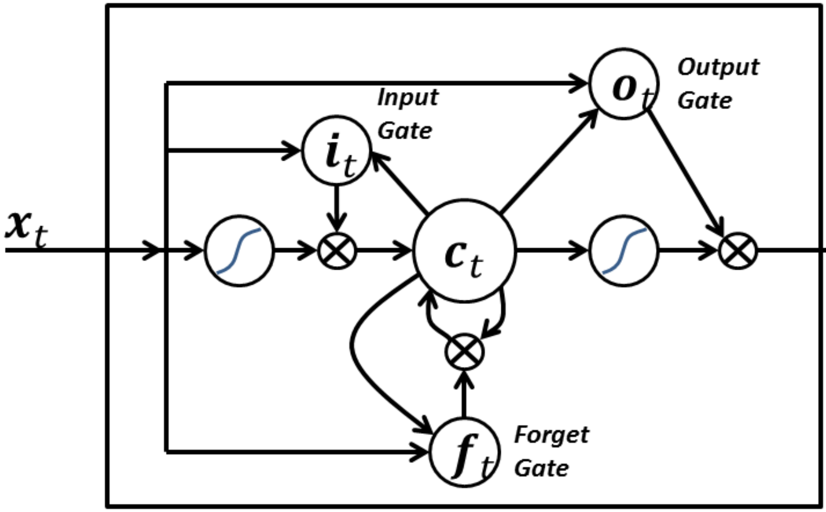
# Long short-term memory (LSTM)

- A more complicated gate, same idea as GRU

- Input gate (current cell matters) $i_t = \sigma\left(W^{(i)}x_t + U^{(i)}h_{t-1}\right)$
- Forget (gate 0, forget past) $f_t = \sigma\left(W^{(f)}x_t + U^{(f)}h_{t-1}\right)$
- Output (how much cell is exposed) $o_t = \sigma\left(W^{(o)}x_t + U^{(o)}h_{t-1}\right)$
- New memory cell $\tilde{c}_t = \tanh\left(W^{(c)}x_t + U^{(c)}h_{t-1}\right)$

Final memory cell: $c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$

Final hidden state: $h_t = o_t \circ \tanh(c_t)$




2 numbers ($c_t$ and $h_t$) represent the state