Gate are way of addressing vanishing gradient

F in lstm is a forget get – computed base on input and previous state

W’s superscript with inputs and outputs



Push thorugh a sigmoid

So we have an ht-1 multipied by f to get ht

So if the value in f is clos to zero then h is zeroed out

If f is around one the it is simply a sum - no vanishing gradient

But it doesn’t process the sentence. – that is done outside the gate

Stabilizes the elman network

LSTM

Use this gate to rember information for a longer period of time

Hidden state is the short term memepry and the gate is long term

We also have acell state



So we feed h both up and forward in the network

Forget gate allow the nextwork to chose what to forget or rember (if f near 1 remember if 0 forget)

Information gate gard the g with is the activation function tanh

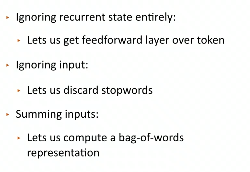
Easier to learn and thread information

So h ends up being a filtered down cell state

Why?

If ignore recurrent we have a feed forward layer

Should be able to do what dan does



Becouse of the way the gate works – there is no longer and exploding or vanishing gradient

GRU is another RNN – simpler to compute but generally doesn’t perform as well

There is an update gat and reset zet gate to combine infor with update

