Module 5

Limitation of vanilla Vanishing and Exploding Gradients

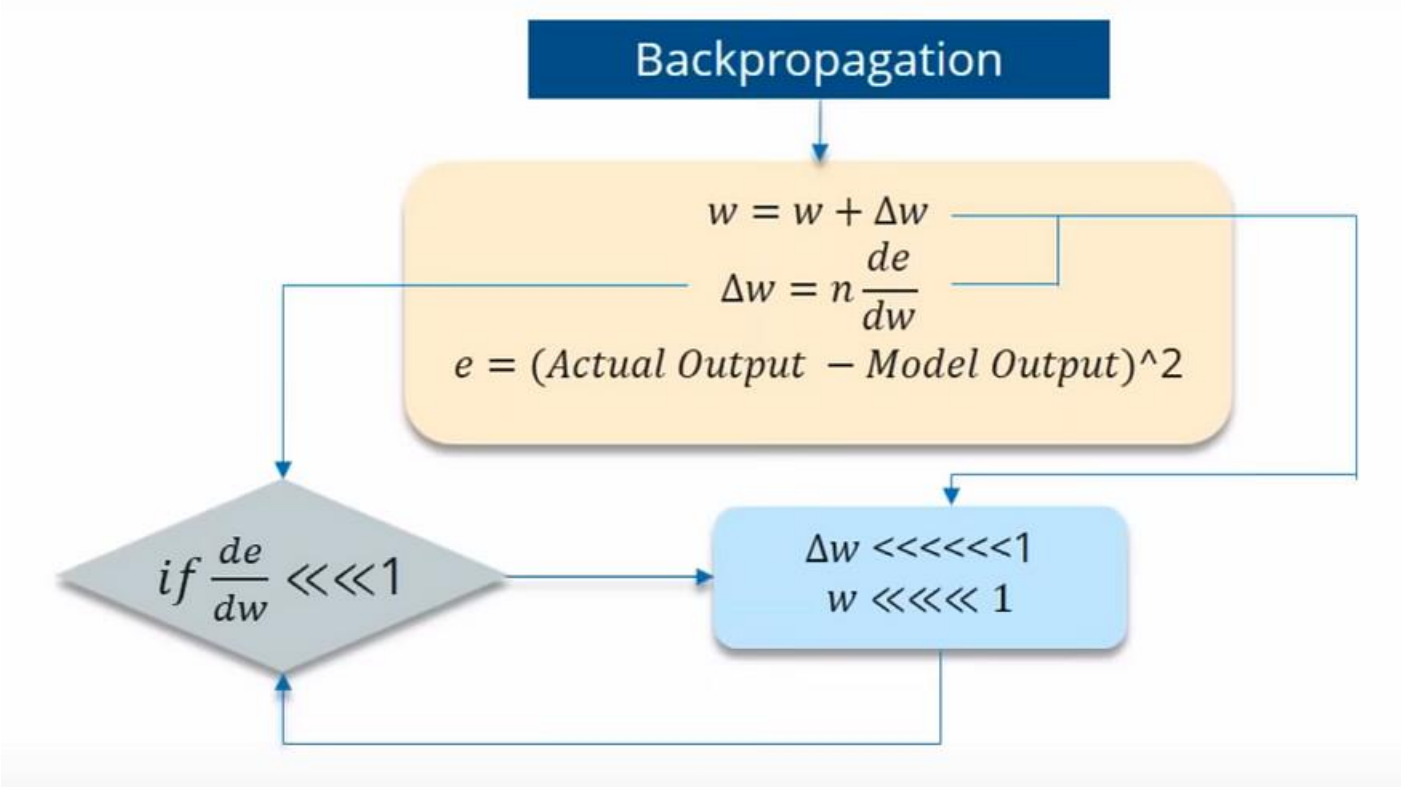
While Backpropogating you may get 2 types of issues.

- **Vanishing Gradient**
- **Exploding Gradient**

Vanishing Gradient:

where the contribution from the earlier steps becomes insignificant in the gradient descent step.

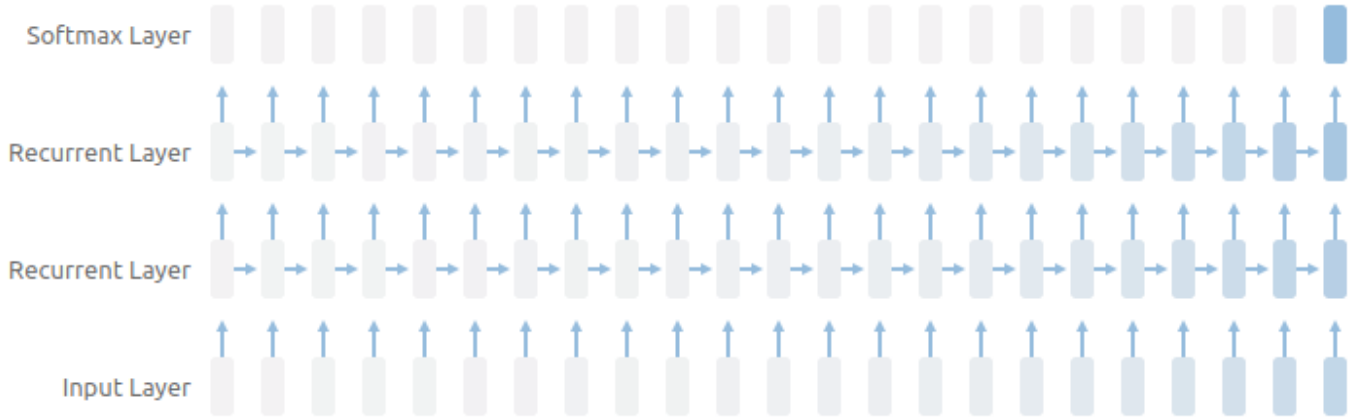
While you are using Backpropogating through time, you find Error is the difference of Actual and Predicted model. Now what if the partial derivation of error with respect to weight is very less than 1?



If the partial derivation of Error is less than 1, then when it get multiplied with the Learning rate which is also very less. then Multiplying learning rate with partial derivation of Error wont be a big change when compared with previous iteration.

For ex:- Lets say the value decreased like 0.863 →0.532 →0.356 →0.192 →0.117 →0.086 →0.023 →0.019..

you can see that there is no much change in last 3 iterations. This Vanishing of Gradience is called **Vanishing Gradience**.



Vanishing Gradient: where the contribution from the earlier steps becomes insignificant in the gradient for the vanilla RNN unit.

Aslo this Vanishing gradient problem results in long-term dependencies being ignored during training.

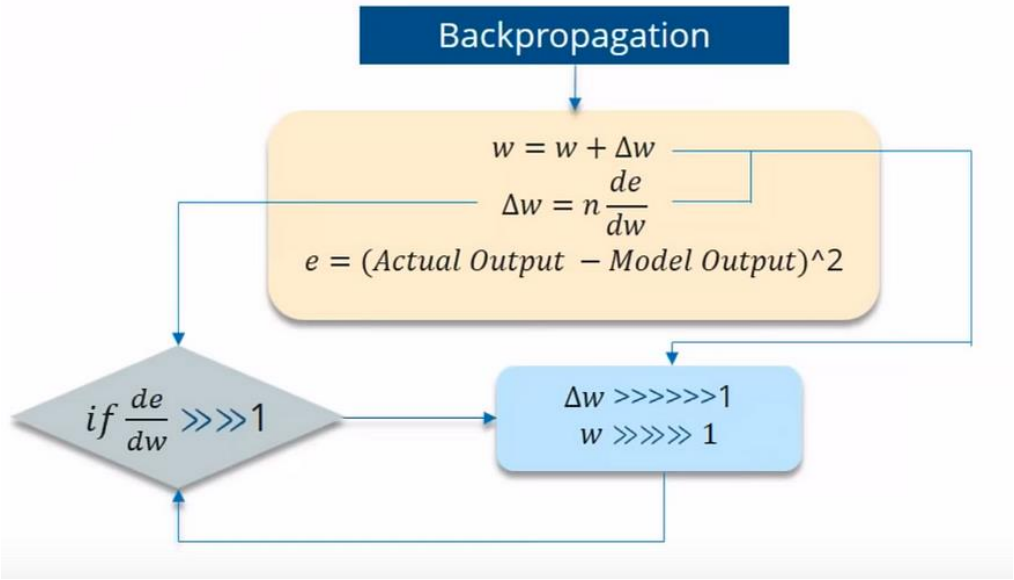
you Can **Visualize** this Vanishing gradient problem at real time **here.**

Several solutions to the vanishing gradient problem have been proposed over the years. The most popular are the aforementioned LSTM and GRU units, but this is still an area of active research.

Exploding Gradient:

We speak of Exploding Gradients when the algorithm assigns a stupidly high importance to the weights, without much reason. But fortunately, this problem can be easily solved if you truncate or squash the gradients.

Module 5



Exploding Gradient

similarly here, What if the Partial derivation of Errror is more than 1? **Think.**

How can you overcome the Challenges of Vanishing and Exploding Gradience?

1. **Vanishing Gradience** can be overcome with

- Relu activation function.
- LSTM, GRU.

2. **Exploding Gradience** can be overcome with

- Truncated BTT(instead starting backprop at the last time stamp, we can choose similar time stamp, which is just before it.)
- Clip Gradience to threshold.
- RMSprop to adjust learning rate.

Advantages of Recurrent Neural Network

1. The main **advantage of RNN** over ANN is that **RNN** can model sequence of data (i.e. time series) so that each sample can be assumed to be dependent on previous ones
2. Recurrent neural network are even used with convolutional layers to extend the effective pixel neighborhood.

Disadvantages of Recurrent Neural Network

Module 5

1. Gradient vanishing and exploding problems.
2. Training an RNN is a very difficult task.
3. It cannot process very long sequences if using *tanh* or *relu* as an activation function.