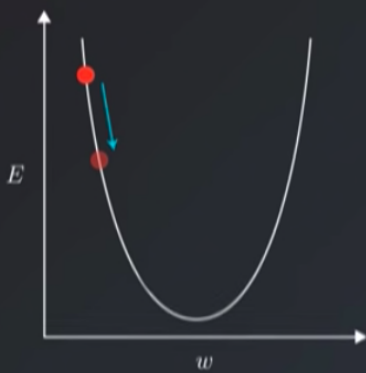


The background is a light gray surface covered with numerous circles of varying sizes and colors. The colors include white, light blue, green, orange, and brown. Some circles are very large, while others are small dots. The circles are scattered across the entire page, creating a playful, confetti-like pattern.

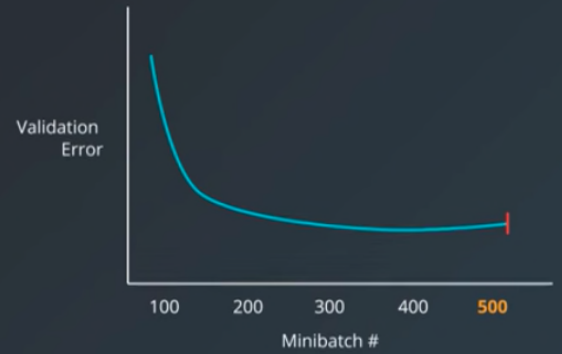
Hyperparameters

Optimizer Hyperparameters



Minitbatch size: 2

x1	x2	x3

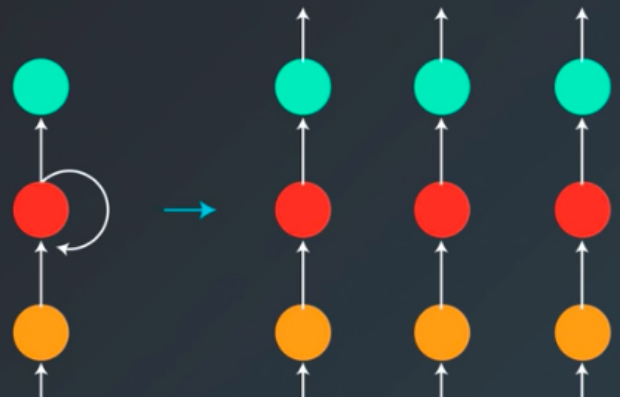


Model Hyperparameters

Input



Output

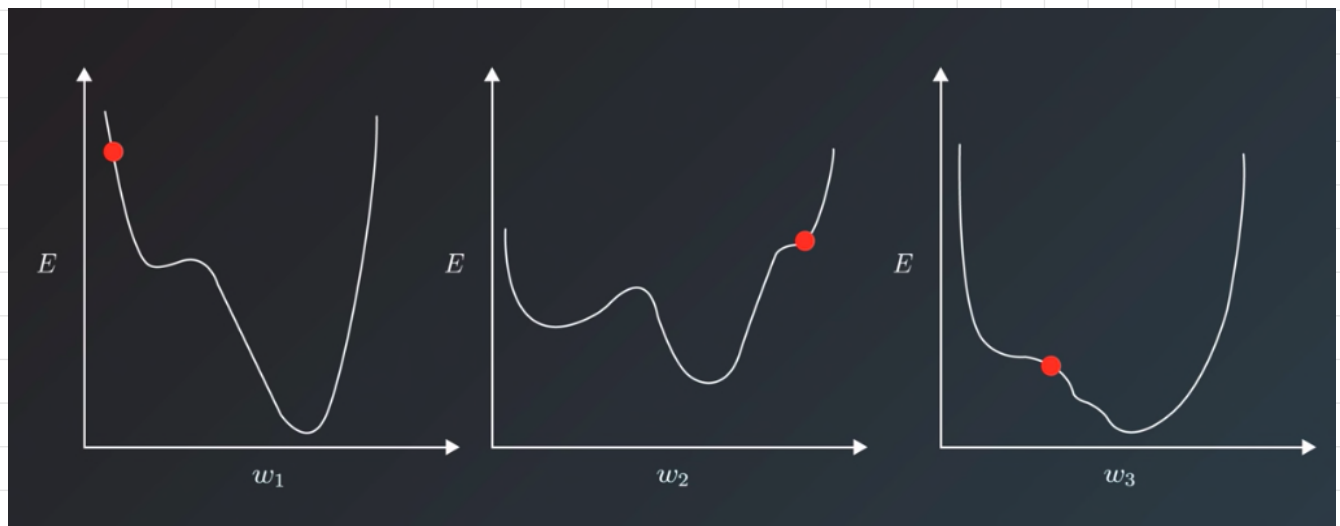


Learning Rate

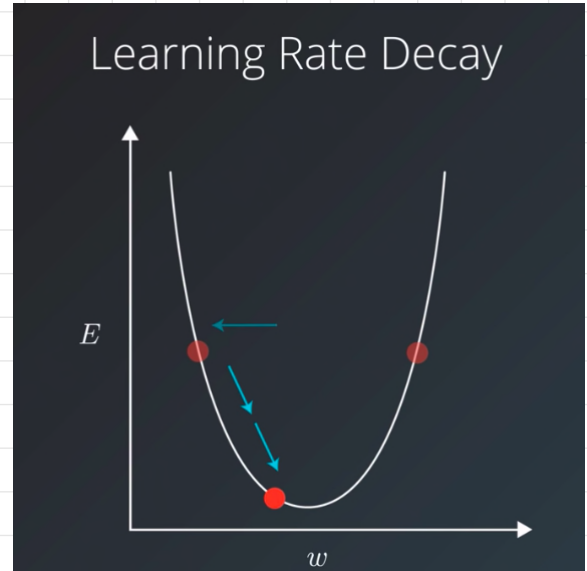
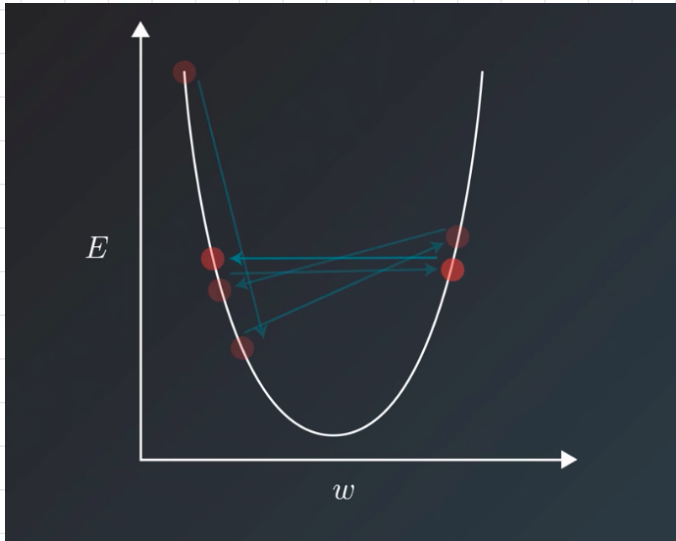
Good start: 0.01



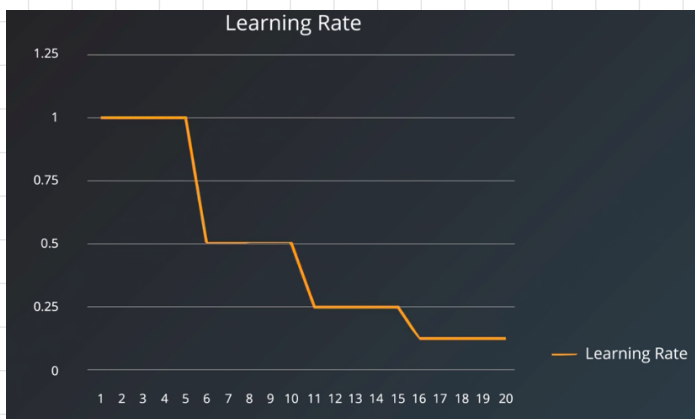
There are, however, much more weights.



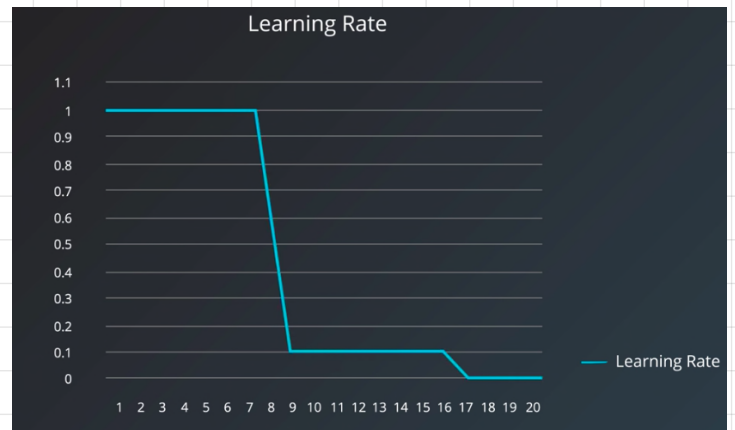
Oscillating loss w/out reaching minimum:



Then it would be useful to employ ↗



Linear



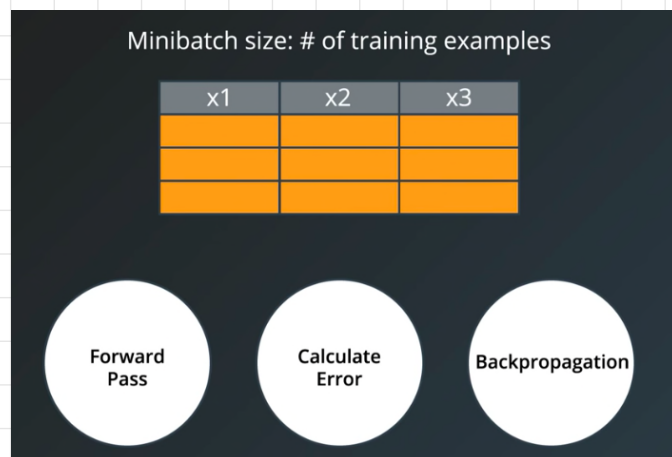
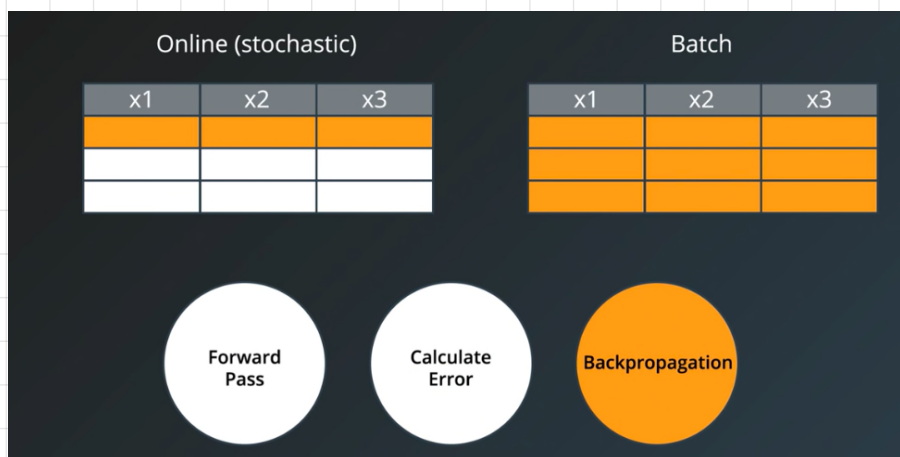
Exponential

OR

Advanced Optimizer

- Adam
- Adagrad

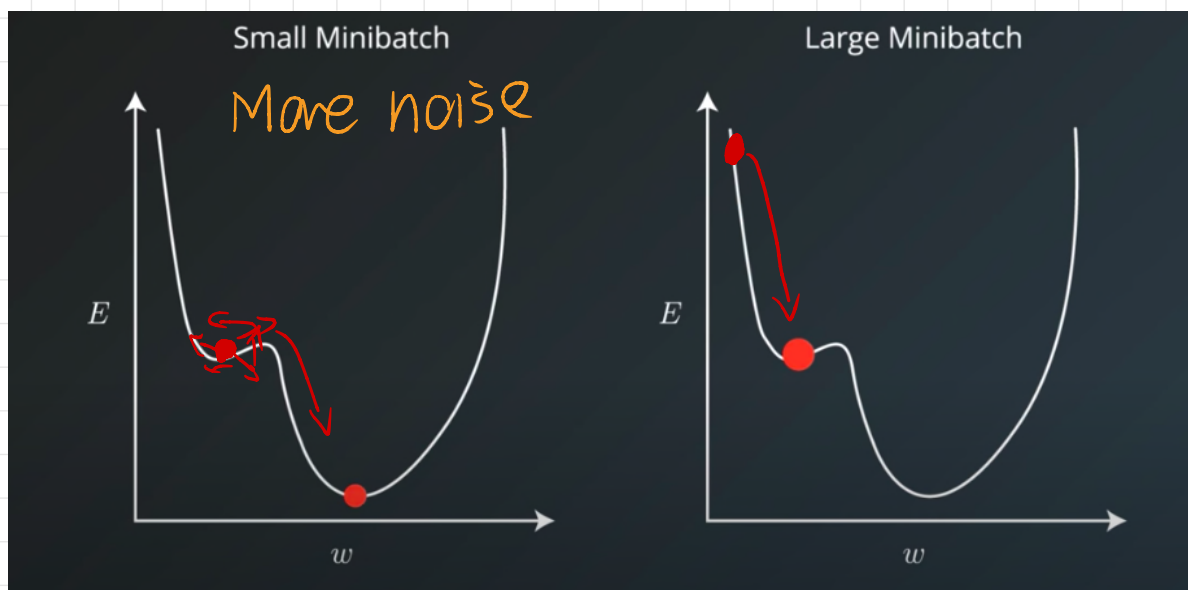
Mini-batch Size



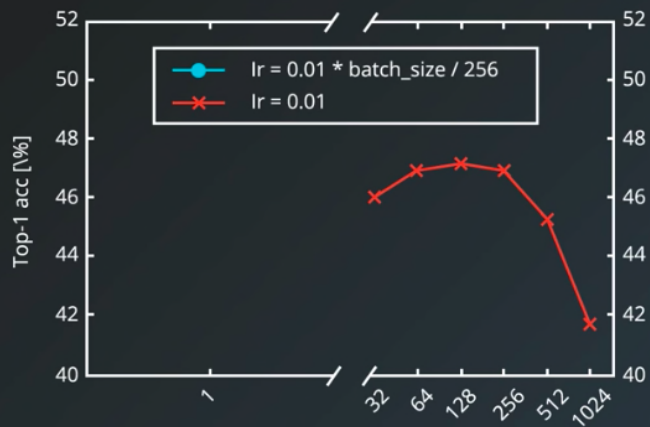
Values to pick: 1, 2, 4, 8, 16, 32, 64...

Good
Start

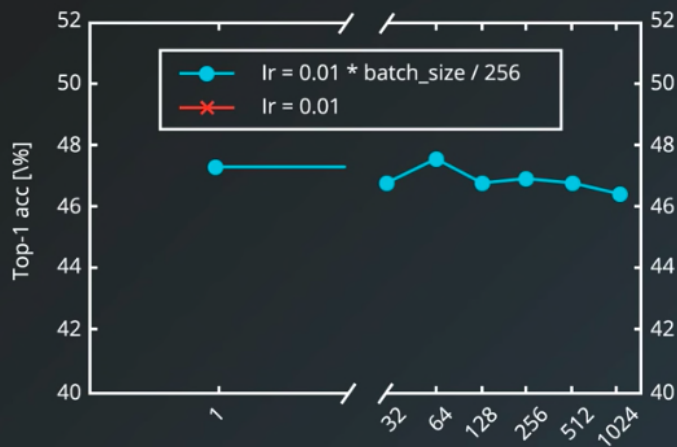
Batch-size ↑ More memory, computation cost.
↓



Noise ↑: can help overcome global minimum!



Source: Mishkin, Dmytro, Nikolay Sergievskiy, and Jiri Matas. "Systematic evaluation of CNN advances on the ImageNet." arXiv preprint arXiv:1606.02228 (2016).

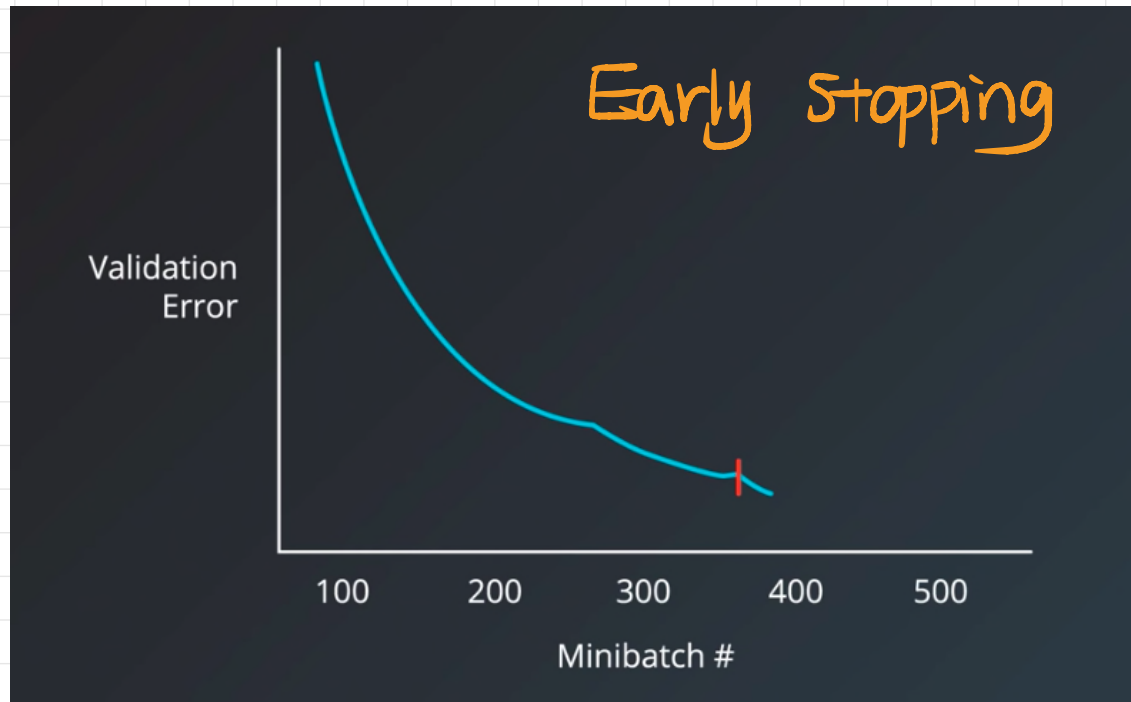


Source: Mishkin, Dmytro, Nikolay Sergievskiy, and Jiri Matas. "Systematic evaluation of CNN advances on the ImageNet." arXiv preprint arXiv:1606.02228 (2016).

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048

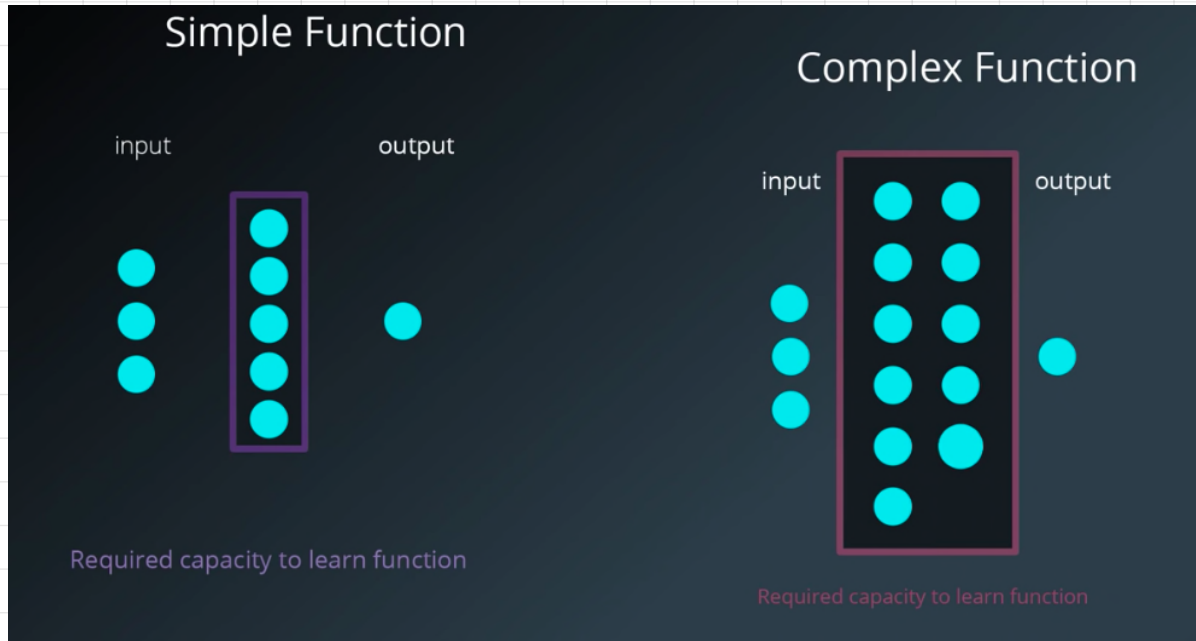
of iterations & Epochs

Metric: validation loss

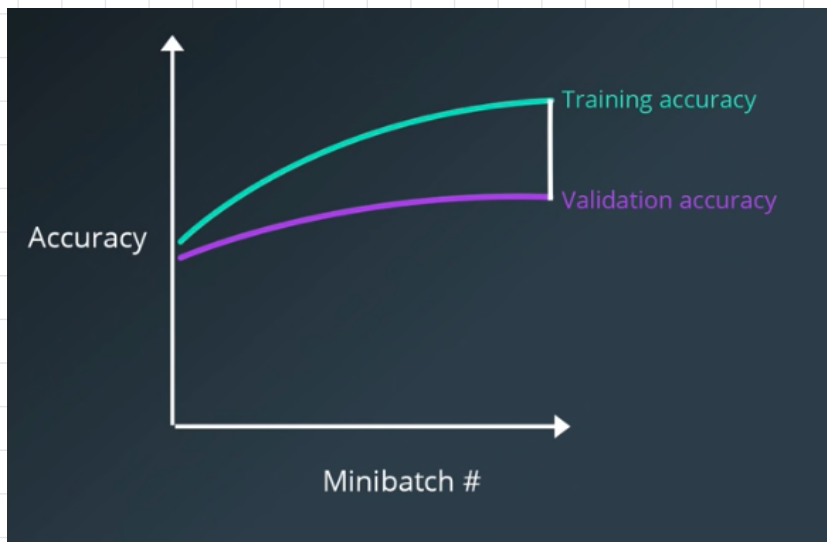


Note: don't stop at when first seeing a bump up of valid loss, do so after few more iterations!

of Hidden Units & Layers



Relationship: learn capacity \propto # of units



Drawback:

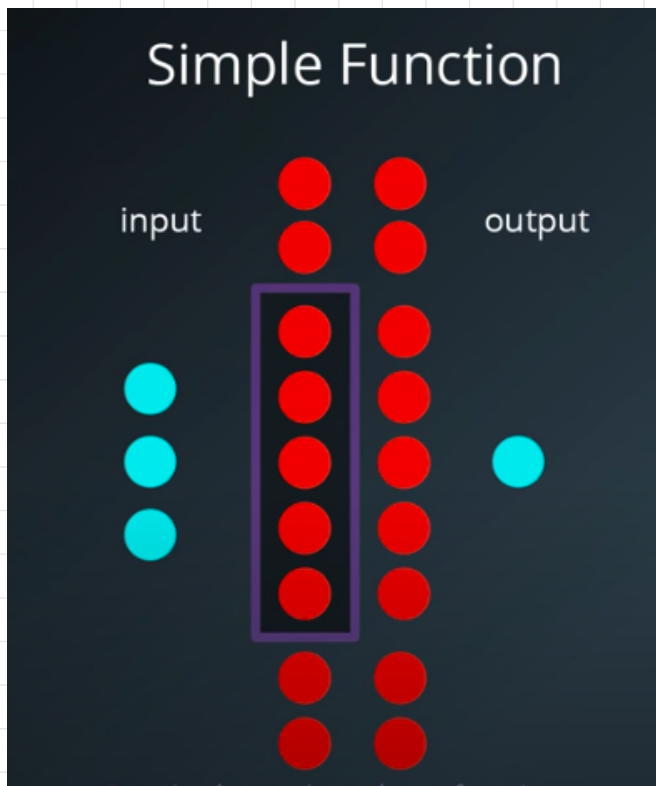
↑ capacity

or

↑ units

leads to **Overfitting!**





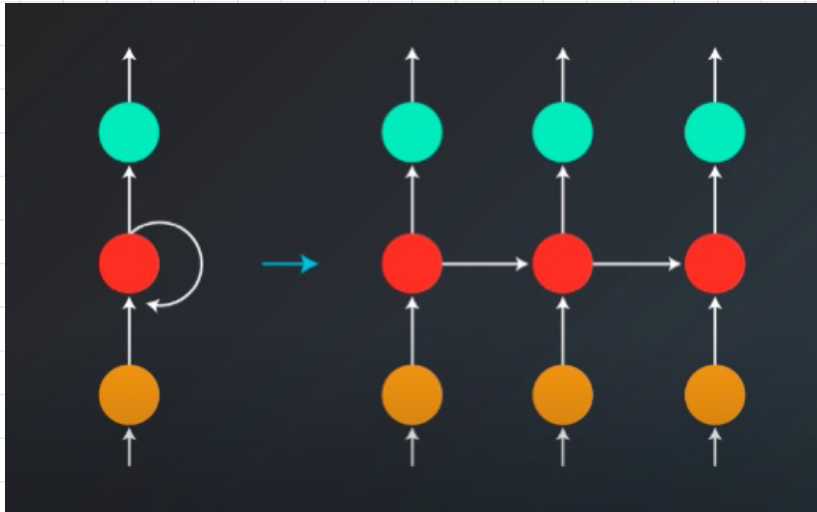
Rule of thumb:

- If model is not learning:
keep adding units till
`valid_loss >> train_loss`
- # of hidden units should
be $>$ # of inputs

Layers?

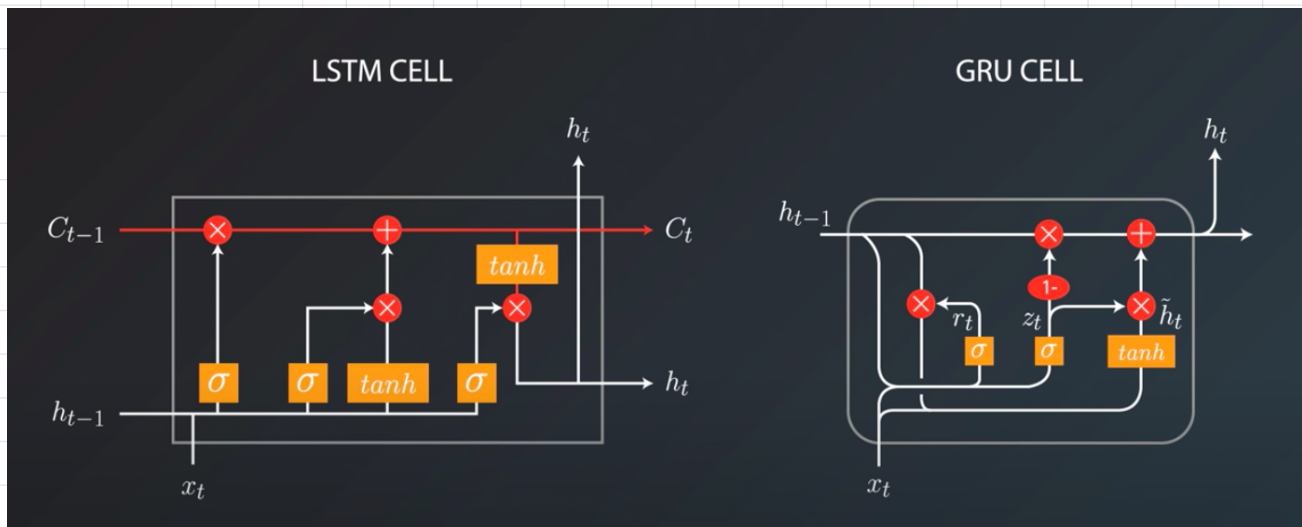
- 3 layers always better than 2
- $>$ 3 layers is a little skeptic
- **Exception:** convolutional NN.
 - The deeper the better!

RNN Hyperparameters

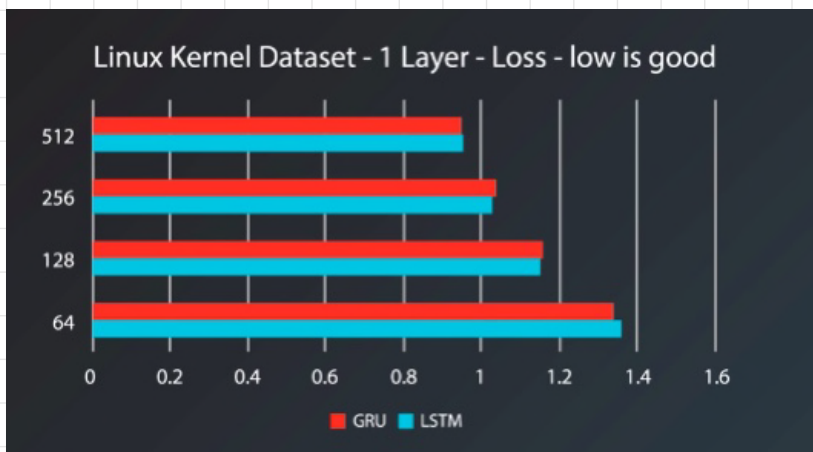


2 Main Questions:

1. Cell type
(LSTM? GRU?)
2. # of layers/stacks



LSTM: more common.



Rule:

- Try both, only on a random subset of the data

