**Selecting Hyperparameters**

1. Number of hidden layers

- one or two hidden layers are sufficient if enough neurons are present

- more layers give higher parameter efficiency: better performance with same amount of data

- increases ability to generate to new data:

- to create new model on the same dataset, may be able to re-use lower levels of first network and their pre-determined weights and biases (called **transfer learning**)

- approach**:** start with one or two, increase until overfitting training data

1. Number of neurons per layer

- input and output numbers are determined by the task required

- common practice to size hidden layers into a pyramid, most in the first and least in the last

- however, it is shown that using the same number of neurons per layer works just as well

- for complex problems, increase neurons until overfitting. However, increasing depth generally works better

3) Learning Rate

- arguably most important

- optimal learning rate: half of maximum learning rate (at which algorithm diverges)

- approach: start with a large value which makes algorithm diverge, divide by 3, and repeat until no longer diverging

4) Batch size

- Should be <32, small batch size ensures each training is fast

- should be >20, helps take advance of soft/hardware optimization, especially for matrix

5) Epochs(number of training we will do)

- use a large number

- then use early-stopping technique to avoid overfitting