How to improve a neural Network

1. Fine tuning neural network hyper parameters
   example number of hidden layers, neurons
   per layer, learning rate, optimizer, Batch size,
   activation function, number of epochs

2. By Solving problems:
   → Vanishing /Exploding gradient
   → Not enough data
   → slow training
   → Overfitting.

Fine Tuning neural network hyper parameters:

1. Number of hidden Layers:
   - three hidden layer with 32 neurons each
     is better from the 1 hidden layer with
     512 neurons.
   - we can increase the hidden layer till
     Overfitting.
   - Because initial layer capture the information
     of lines and shapes. example transfer
     learning in CNN

2. Neurons per layer:
   - Number of neurons in input layer is number
     of input data
   - Number of neurons in output layer is based
     one problem that we solve. example
     1 in regression.

- Number of neuron hidden layer. is in sufficient number to capture the. all premitive features.

3. Batch size :
- select smaller batch size (8 to 32) because get better results on new data but it is slower. execution.
- select larger batch size (8 92) because here the execution is fast or an another way we can use learning rate schedular. if we use the larger batch size.
- (learning rate increase from smaller to larger) we call it as. warming up the learning. rate.

4. Epochs size:
- Here we use. the mechanism early stopping. in keras.
- it is automatically detect in which phase of. ~~epok~~ epochs. we stop the execution.

Solving Problem with Neural Network:

1. Vanishing / exploding gradients:
- weight initialisation
- ~~Actio~~ Activation function selection
- Batch. Normalisation
- Gradient clipping

2. Not enough Data:
- use transfer learning
- unsupervised pre-training

3. Slow training:
- use many type of optimizer example adam
- use learning rate scheduler.

4. Overfitting:
- use L1 and L2 regularisation
- Dropout.

Improve Neural Networks Performance:

1. Vanishing Gradients.
- Activation functions
- weight initialization.

2. Overfitting.
- Reduce Complexity / Increase Data
- Dropout layer.
- Regularization (L1 and L2)
- Early stopping

3. Normalization
- Normalizing inputs
- Batch Normalization
- Normalizing Activations

4. Gradient Checking and clipping

5. Optimizers
- Momentum
- Adagrad

- RMSSprop
- Adam

6. Learning Rate Scheduling

7. Hyperparameter tunning
   - Number of hidden Layers
   - Nodes/layer
   - Batch size