1. How can each of these parameters be fine-tuned?

• Number of hidden layers

The number of hidden layers can be adjusted based on the complexity of the problem. Increasing the number of layers allows the network to learn more intricate representations, but too many layers can lead to over fitting. It often requires empirical experimentation to determine the optimal number of hidden layers for a specific task.

• Network architecture (network depth)

The network architecture, including the depth and structure of the layers, can be adjusted to optimize model performance. This involves experimenting with different architectures, such as adding or removing layers, altering skip connections, or using specialized architectures like residual networks or convolutional architectures for specific tasks like image processing.

• Each layer's number of neurons (layer width)

The number of neurons in each layer, also known as layer width, can impact the model's capacity and learning capability. Increasing the number of neurons can enhance the model's ability to learn complex patterns, but it may also lead to over fitting

• Form of activation

The choice of activation functions can greatly influence the model's ability to capture non-linear relationships and learn complex representations. Experimenting with different activation functions, such as ReLU, sigmoid, or tanh,

• Optimization and learning

Selecting the appropriate optimization algorithm, such as stochastic gradient descent (SGD), Adam, or RMSprop, impacts the convergence speed and final performance of the model. Additionally, adjusting hyperparameters related to the optimization process, such as momentum, weight decay, or adaptive learning rates, can further fine-tune the training process.

• Learning rate and decay schedule

The learning rate controls the step size during parameter updates. Tuning the learning rate and utilizing learning rate schedules or decay mechanisms can ensure a balance between fast initial learning and more precise fine-tuning.

• Mini batch size

The learning rate controls the step size during parameter updates. Tuning the learning rate and utilizing learning rate schedules or decay mechanisms can ensure a balance between fast initial learning and more precise fine-tuning.

• Algorithms for optimization

Algorithms for optimization commonly used in machine learning and deep learning. Here are some popular optimization algorithms: Stochastic Gradient Descent (SGD), Mini-Batch Gradient Descent, Adam (Adaptive Moment Estimation) and RMSprop (Root Mean Square Propagation).

• The number of epochs (and early stopping criteria)

The number of epochs refers to the number of times the entire training dataset is passed forward and backward through the neural network during the training process. It determines the number of iterations the model undergoes to update its parameters based on the training data.

• Overfitting that be avoided by using regularization techniques.

Over fitting occurs when a machine learning model learns to perform well on the training data but fails to generalize effectively to unseen data. Regularization techniques are employed to prevent over fitting by adding constraints to the model's learning process. These techniques help to control the model's complexity, reduce its sensitivity to noisy or irrelevant patterns in the training data, and encourage better generalization

• L2 normalization

L2 normalization, also known as weight decay or L2 regularization, is a regularization technique used in machine learning and deep learning to mitigate over fitting and improve the generalization ability of models. It adds a penalty term to the loss function during training, discouraging the model from assigning excessively large weights to the parameters.

• Drop out layers

Dropout is a popular regularization technique used in neural networks to combat over fitting and improve generalization. Dropout layers are inserted into the network architecture and randomly deactivate a fraction of neurons during training.

• Data augmentation

Data augmentation techniques, such as random rotations, translations, or flips, can artificially increase the size of the training dataset and improve the model's ability to generalize by exposing it to diverse variations of the input data.