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

1. Number of hidden layers
2. Network architecture (network depth)
3. Each layer's number of neurons (layer width)
4. Form of activation
5. Optimization and learning
6. Learning rate and decay schedule
7. Mini batch size
8. Algorithms for optimization
9. The number of epochs (and early stopping criteria)
10. Overfitting that be avoided by using regularization techniques.
11. L2 normalization
12. Drop out layers
13. Data augmentation

Answer:

1. Number of hidden layers: The number of hidden layers can be fine-tuned by testing different architectures and observing their performance on the validation set.
2. Network architecture (network depth): The network architecture can be fine-tuned by trying out different types of architectures such as convolutional neural networks, recurrent neural networks, or transformer-based architectures. The depth of the network can also be adjusted based on the complexity of the problem.
3. Each layer's number of neurons (layer width): The number of neurons in each layer can be fine-tuned by testing different values and observing the performance on the validation set. This is often done by using a grid search or random search over a range of values.
4. Form of activation: The form of activation can be fine-tuned by trying out different activation functions such as ReLU, sigmoid, or tanh and observing the performance on the validation set.
5. Optimization and learning: The optimization algorithm and learning rate can be fine-tuned by trying out different optimization algorithms such as stochastic gradient descent, Adam, or RMSProp and adjusting the learning rate to find the best performance on the validation set.
6. Learning rate and decay schedule: The learning rate and decay schedule can be fine-tuned by testing different values of learning rate and observing the performance on the validation set. Different decay schedules such as step decay, exponential decay, or cosine decay can also be tried out.
7. Mini batch size: The mini batch size can be fine-tuned by trying out different batch sizes and observing the performance on the validation set.
8. Algorithms for optimization: Different optimization algorithms such as stochastic gradient descent, Adam, or RMSProp can be fine-tuned by adjusting their hyperparameters and observing the performance on the validation set.
9. The number of epochs (and early stopping criteria): The number of epochs can be fine-tuned by trying out different values and observing the performance on the validation set. Early stopping criteria can also be used to prevent overfitting and fine-tune the number of epochs.
10. Overfitting that can be avoided by using regularization techniques: Regularization techniques such as L2 normalization or dropout layers can be fine-tuned by adjusting their hyperparameters and observing the performance on the validation set.
11. L2 normalization: The L2 normalization parameter can be fine-tuned by trying out different values and observing the performance on the validation set.
12. Drop out layers: The dropout rate can be fine-tuned by trying out different values and observing the performance on the validation set.
13. Data augmentation: The type and degree of data augmentation can be fine-tuned by trying out different techniques such as rotation, translation, or flipping and observing the performance on the validation set.