1)how can each of these parameters be fine-tuned ?

.number of hidden layers

Number of Neurons and Number of Layers in Hidden Layer

The number of hidden neurons should be between the size of the input layer and the size of the output layer. The number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer.product of the number of neurons in the input layer and first hidden layer. sum of products of the number of neurons between the two consecutive hidden layers.

.optimization & learning.

fancy name for training: the selection of parameter values, which are optimal in some desired sense (eg. minimize an objective function you choose over a dataset you choose). The parameters are the weights and biases of the network.

Optimized parameter values will enable the model to perform the task with relative accuracy. The cost function inputs a set of parameters and outputs a cost, measuring how well that set of parameters performs the task (on the training set).

.learning rate & decay schedule

The learning rate is the most important hyper-parameter for tuning neural networks. A good learning rate could be the difference between a model that doesn't learn anything and a model that presents state-of-the-art results.The learning rate in fine-tune CNN layers is set to 0.0001, while the learning rate in fine-tune BN layers is set into 0.01. We found that such a learning rate setting can make the network obtain better results.

The mathematical form of time-based decay is lr = lr0/(1+kt) where lr , k are hyperparameters and t is the iteration number. Looking into the source code of Keras, the SGD optimizer takes decay and lr arguments and update the learning rate by a decreasing factor in each epoch.

.mini batch size

Batch size is one of the most important hyperparameters to tune in modern deep learning systems. Practitioners often want to use a larger batch size to train their model as it allows computational speedups from the parallelism of GPUs.The best performance has been consistently obtained for mini-batch sizes between 2 and 32. This contrasts with recent work, which is motivated by trying to induce more data parallelism to reduce training time on today's hardware.Our results concluded that a higher batch size does not usually achieve high accuracy, and the learning rate and the optimizer used will have a significant impact as well. Lowering the learning rate and decreasing the batch size will allow the network to train better, especially in the case of fine-tuning.

.Algorithm for optimization.

fancy name for training: the selection of parameter values, which are optimal in some desired sense (eg. minimize an objective function you choose over a dataset you choose). The parameters are the weights and biases of the network.

SGD is the most important optimization algorithm in Machine Learning. Mostly, it is used in Logistic Regression and Linear Regression.Validation set is used for tuning the parameters of a model.fine-tuning is the process in which parameters of a model must be adjusted very precisely in order to fit with certain observations.

.The number of epochs(and early stopping Criteria)

This strategy of stopping early based on the validation set performance is called Early Stopping. This is explained with the below diagram. The validation set accuracy, however, saturates between 8 to 10 epochs. This is where the model can be stopped training.There are three elements to using early stopping; they are: Monitoring model performance. Trigger to stop training. The choice of model to use.

Increasing epochs makes sense only if you have a lot of data in your dataset. However, your model will eventually reach a point where increasing epochs will not improve accuracy.The right number of epochs depends on the inherent perplexity (or complexity) of your dataset. A good rule of thumb is to start with a value that is 3 times the number of columns in your data. If you find that the model is still improving after all epochs complete,

.overfitting that be avoided by using Regularization Techniques.

It is one of the most important concepts of machine learning. This technique prevents the model from overfitting by adding extra information to it.

It is a form of regression that shrinks the coefficient estimates towards zero. In other words, this technique forces us not to learn a more complex or flexible model, to avoid the problem of overfitting.In the Regularization technique, we reduce the magnitude of the independent variables by keeping the same number of variables”. It maintains accuracy as well as a generalization of the model.

Regularization works by adding a penalty or complexity term or shrinkage term with Residual Sum of Squares (RSS) to the complex model.

Let’s consider the Simple linear regression equation:

Here Y represents the dependent feature or response which is the learned relation. Then,

Y is approximated to β0 + β1X1 + β2X2 + …+ βpXp

Here, X1, X2, …Xp are the independent features or predictors for Y, and

β0, β1,…..βn represents the coefficients estimates for different variables or predictors(X), which describes the weights or magnitude attached to the features, respectively.

In simple linear regression, our optimization function or loss function is known as the residual sum of squares (RSS).

We choose those set of coefficients, such that the following loss function is minimized:

Regularization

Fig. Cost Function For Simple Linear Regression

.L2 Normalization.

L2 regularization term is the sum of squared values of each element. For a length N vector, it would be w[1]² + w[2]² + ... + w[N]² . I hope this helps L1 Regularization, also called a lasso regression, adds the “absolute value of magnitude” of the coefficient as a penalty term to the loss function. L2 Regularization, also called a ridge regression, adds the “squared magnitude” of the coefficient as the penalty term to the loss function.

Dropout layer

good rule of thumb is to divide the number of nodes in the layer before dropout by the proposed dropout rate and use that as the number of nodes in the new network that uses dropout. For example, a network with 100 nodes and a proposed dropout rate of 0.5 will require 200 nodes (100 / 0.5) when using dropout.This is in contrast to setting trainable=False for a Dropout layer. trainable does not affect the layer's behavior, as Dropout does not have any variables/weights that can be frozen during training.)

.Data augmentation.

Some of the most common data augmentation techniques used for images are:

Position augmentation. Scaling. Cropping. Flipping. Padding. Rotation. Translation. Affine transformation.

Color augmentation. Brightness. Contrast. Saturation. Hue.One of the best techniques for reducing overfitting is to increase the size of the training dataset. As discussed in the previous technique, when the size of the training data is small, then the network tends to have greater control over the training data.