Hyperparameter tuning of Deep learning models in Keras

KEYWORDS: Deep Learning, Model tuning, Hyperparameters, Hyperparameter tuning, Keras, Keras tuner, AiSara tuner, machine learning tuning

Reading time: 10 mins approx.

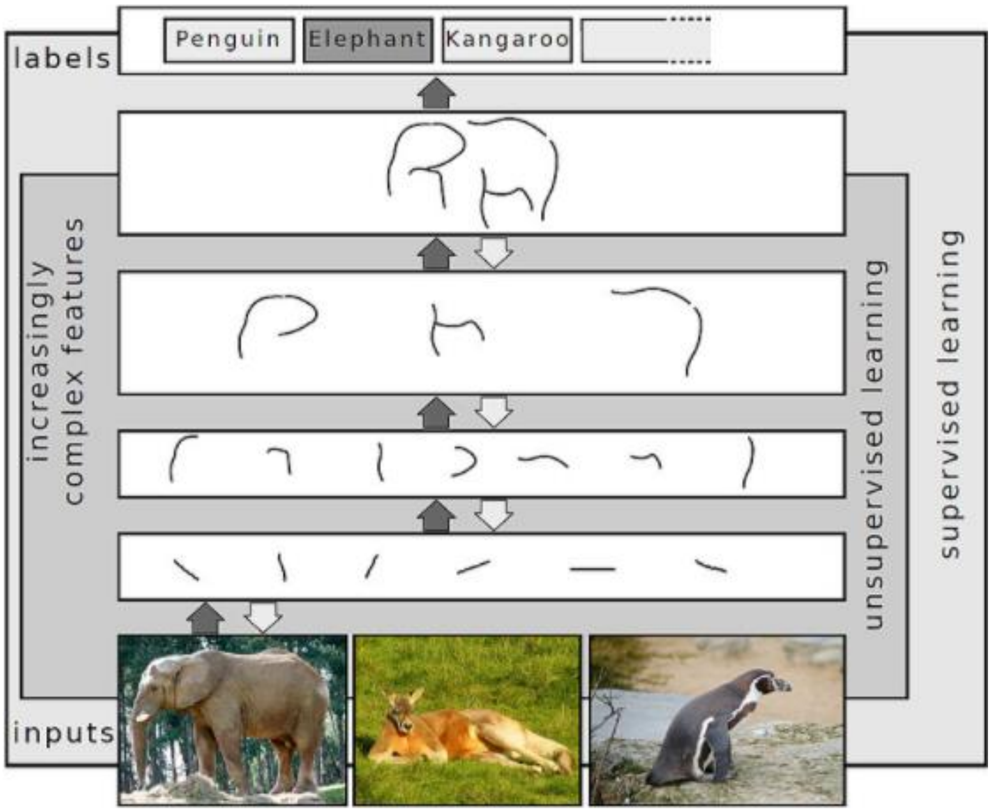


Fig: Representing Images on Multiple Layers of Abstraction in Deep Learning

Source: https://en.wikipedia.org/wiki/Deep_learning

Machine learning model tuning is usually a trial-and-error process through which we change some hyperparameters, run the model on the data again with the objective of improving the performance to determine which set of hyperparameters results in the most accurate model.

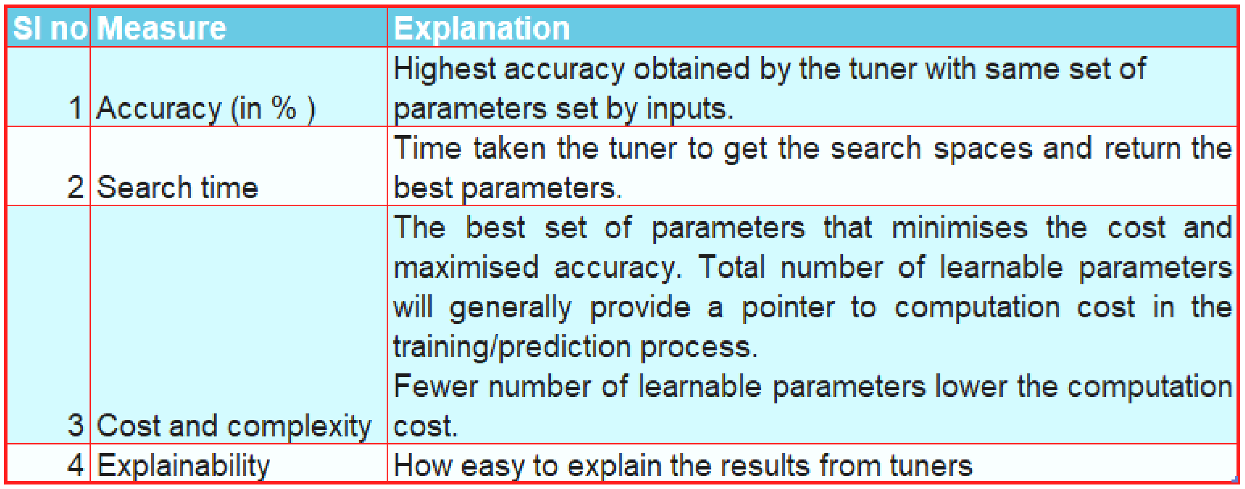
In my persuit of building efficient deep learning models, I have stumbled upon at model tuning stage of getting the best accuracy.

I learnt that accuracy-complexity balance depend on many hyperparameters and value of those parameters will have wide range. So the challenge became clear. There is no single solution that can give me one best solution. It involved several trials with hyperparameters and range of values they can accept.

So I decided to master the science of tuning of deep learning models. At first, I found a tuning approach using Keras Tuner. The example python script found in google colab was very useful. Soon I realized I have to go beyond Keras tuner as my goal was to design an efficient deep learning model. I have found another tool called AiSara for tuning deep learning model.

I have experimented with both the Keras tuners with various hyperparameters and benchmarked some measures to compare the tools. I have used fashion MNIST dataset and CIFAR-10 datasets for my experiments as it is convenient for anyyone who want to evaluate the tools from anywhere.

Metrics used for benchmark:



Sl no	Measure	Explanation
1	Accuracy (in %)	Highest accuracy obtained by the tuner with same set of parameters set by inputs.
2	Search time	Time taken the tuner to get the search spaces and return the best parameters.
3	Cost and complexity	The best set of parameters that minimises the cost and maximised accuracy. Total number of learnable parameters will generally provide a pointer to computation cost in the training/prediction process. Fewer number of learnable parameters lower the computation cost.
4	Explainability	How easy to explain the results from tuners

Experimental setup:

I have chosen a basic experimental dataset to enable all reviewers and learners to easily understand and verify the results by using commonly available dataset. So the scripts provided here can be executed from Google Colab by anyone from anywhere. All the experiments are run in Google Colab and shared in my GitHub repo here.

Physical environment

I have set Google Colab with Runtime hardware accelearator = "GPU" (Menu command Navigator: Runtime → Change runtime type)

Dataset

fashion MNIST dataset

```
import tensorflow as tf
from tensorflow import keras
```

```
# load fashion_mnist dataset
(img_train, y_train), (img_test, y_test) = keras.datasets.fashion_mnist.load_data()
```

CIFAR-10 dataset

```
(img_train, y_train), (img_test, y_test) = keras.datasets.cifar10.load_data()
```

Keras tuners

I have separately set up mirror environment to Keras tuner and AiSara tuner in separate python scripts.

Keras tuner

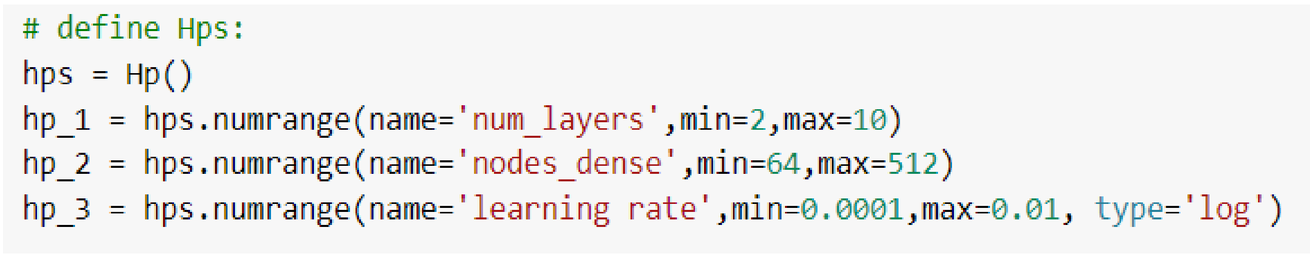
```
!pip install -q -U keras-tuner
import kerastuner as kt
```

AiSara tuner

```
!pip install aisaratuners
from aisaratuners.aisara_keras_tuner import Hp, HpOptimization
```

Hyperparameters:

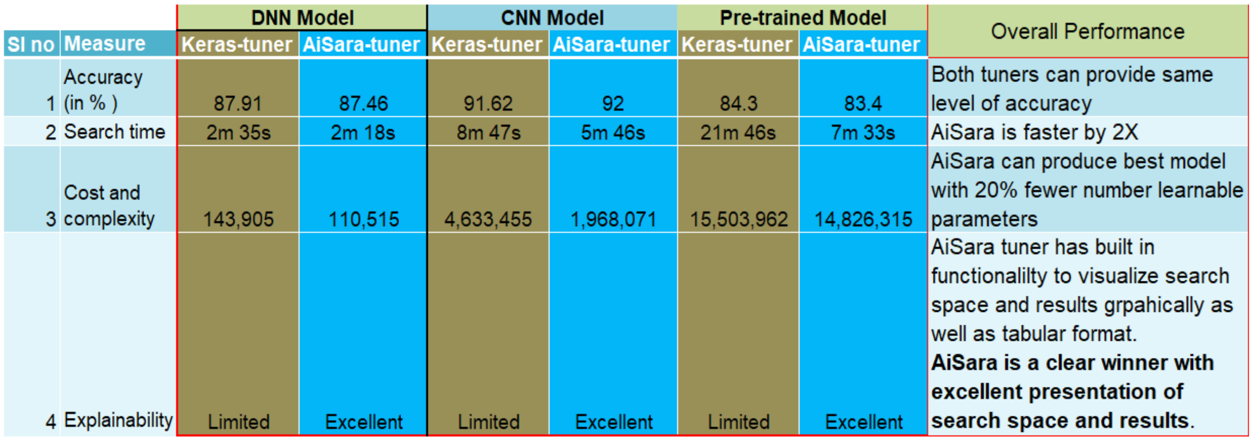
I have used 3 hyperparameters number of layers, Number of units, learning rate in my experiments.



```
# define Hps:
hps = Hp()
hp_1 = hps.numrange(name='num_layers',min=2,max=10)
hp_2 = hps.numrange(name='nodes_dense',min=64,max=512)
hp_3 = hps.numrange(name='learning rate',min=0.0001,max=0.01, type='log')
```

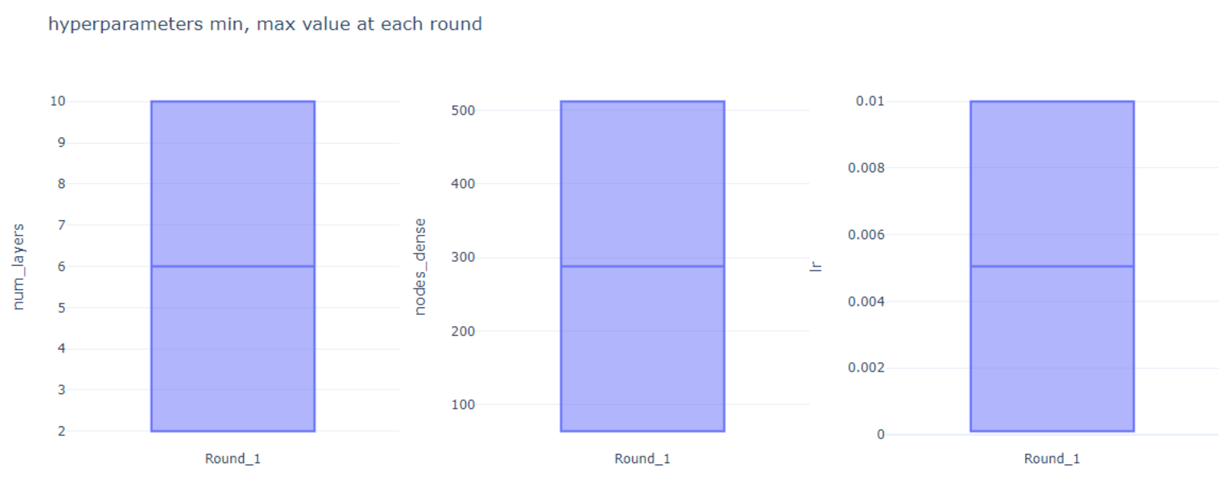
Experimental Results

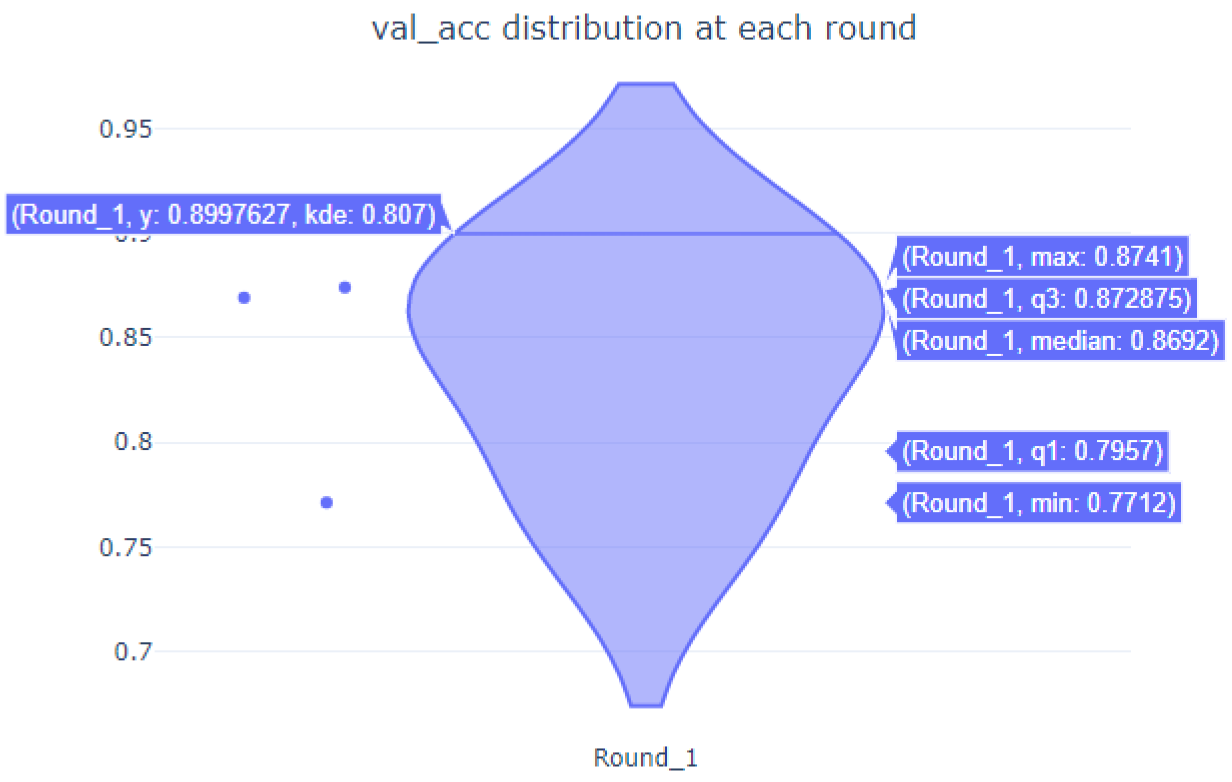
Below is the summary of all experimental results. You can refer to GitHub repo here to view the source and results obtained from test runs.



		DNN Model		CNN Model		Pre-trained Model		Overall Performance
Sl no	Measure	Keras-tuner	AiSara-tuner	Keras-tuner	AiSara-tuner	Keras-tuner	AiSara-tuner	
1	Accuracy (in %)	87.91	87.46	91.62	92	84.3	83.4	Both tuners can provide same level of accuracy
2	Search time	2m 35s	2m 18s	8m 47s	5m 46s	21m 46s	7m 33s	AiSara is faster by 2X
3	Cost and complexity	143,905	110,515	4,633,455	1,968,071	15,503,962	14,826,315	AiSara can produce best model with 20% fewer number learnable parameters
4	Explainability	Limited	Excellent	Limited	Excellent	Limited	Excellent	AiSara tuner has built in functionalilty to visualize search space and results grpahically as well as tabular format. **AiSara is a clear winner with excellent presentation of search space and results.**

AiSara Keras tuner has excellent graphical visualization methods which can explain the search space. Few samples of visualizations provided by AiSara keras tuner shown below:





I hope this experiment will help all data science enthusiasts and deep learning buddies to understand the keras tuners.