Neural Networks hyperparameters

Damian Podareanu

SURFsara

Parameters vs. Hyperparameters

Parameters: weights (W_1, W_2, ...) and biases (b_1, b_2, ...)

Hyperparameters related to network design

- number of hidden layers L
- number of hidden units n_1, n_2, …
- dropout (regularization)
- weight initialization
- activation function – tanh, ReLU, …

Hyperparameters for the optimization process

- learning rate α
- number of iterations for gradient descent
- momentum
- minibatch size
- number of epochs

Getting the correct number of neurons on layers

First layer activation

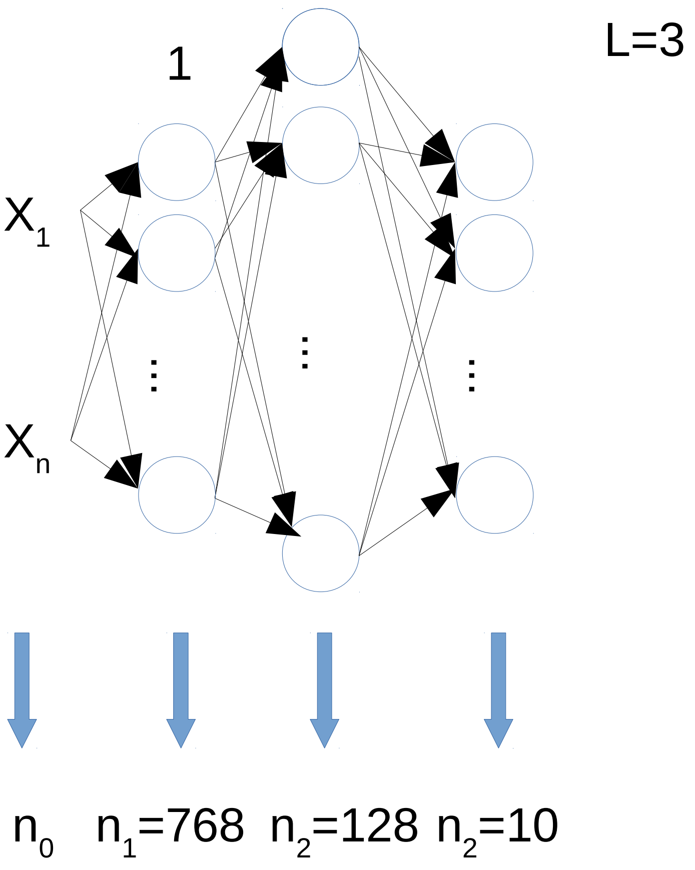
$Z_1 = W_1 * X + b_1$

So in terms of dimensionality:

$(n_1, 1) = (?, ?) * (n_0,1) + (n_1, 1)$

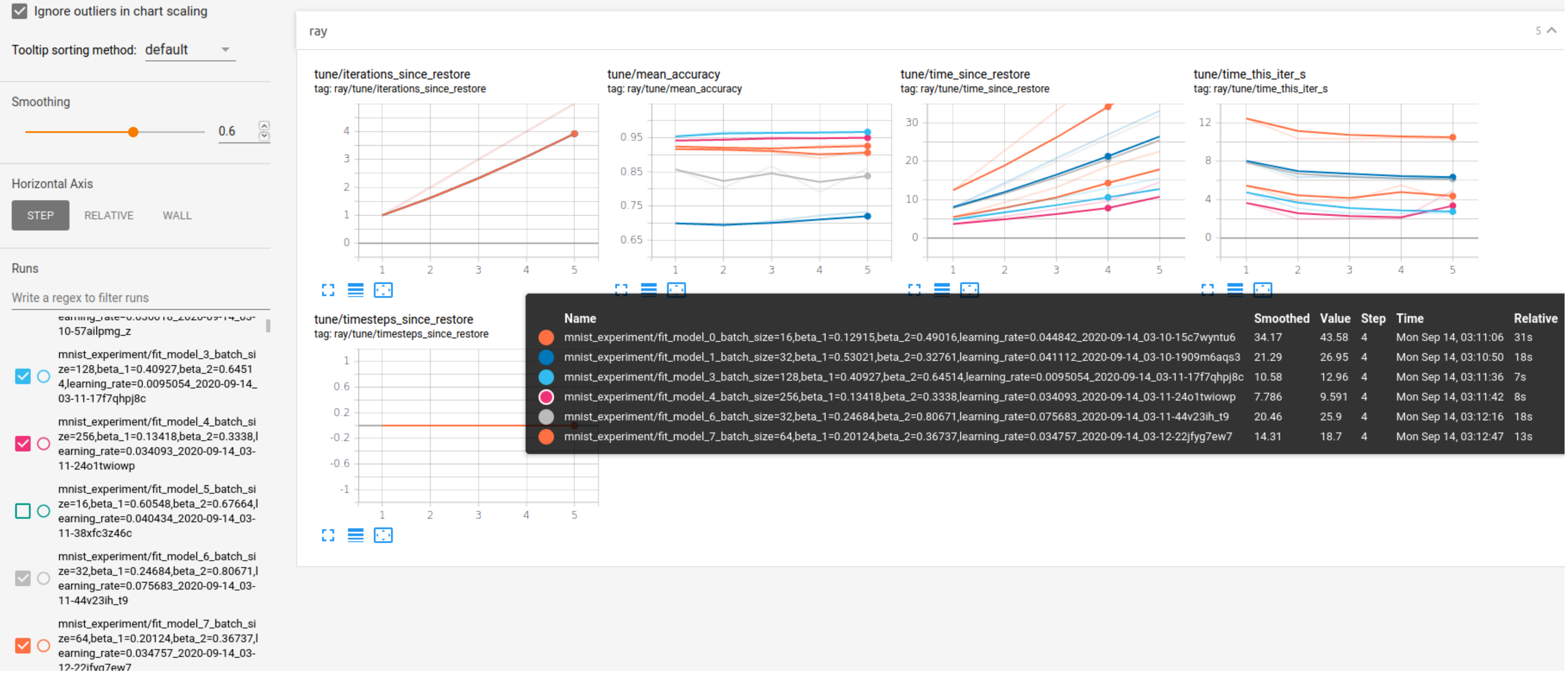
$\rightarrow W_1 : (n_1, n_0)$

$\rightarrow b_1 : (n_1,1)$



Empirical tuning of parameters – Building intuition

Idea → Experiment → Code and repeat


ray
5
tune/iterations_since_restore
tag: ray/tune/iterations_since_restore
tune/mean_accuracy
tag: ray/tune/mean_accuracy
tune/time_since_restore
tag: ray/tune/time_since_restore
tune/time_this_iter_s
tag: ray/tune/time_this_iter_s
tune/timesteps_since_restore
tag: ray/tune/timesteps_since_restore
0
1
2
3
4
5
0.65
0.75
0.85
0.95
10
20
30
8
12
-1
-0.6
-0.2
0.2
0.6

☑ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing 0.6

Horizontal Axis: [✓ STEP] RELATIVE WALL

Runs

Write a regex to filter runs

10-57ailpmg_z

☑ ○ mnist_experiment/fit_model_3_batch_size=128,beta_1=0.40927,beta_2=0.64514,learning_rate=0.0095054_2020-09-14_03-11-17f7qhpj8c

☑ ○ mnist_experiment/fit_model_4_batch_size=256,beta_1=0.13418,beta_2=0.3338,learning_rate=0.034093_2020-09-14_03-11-24o1twiowp

☐ ○ mnist_experiment/fit_model_5_batch_size=16,beta_1=0.60548,beta_2=0.67664,learning_rate=0.040434_2020-09-14_03-11-38xfc3z46c

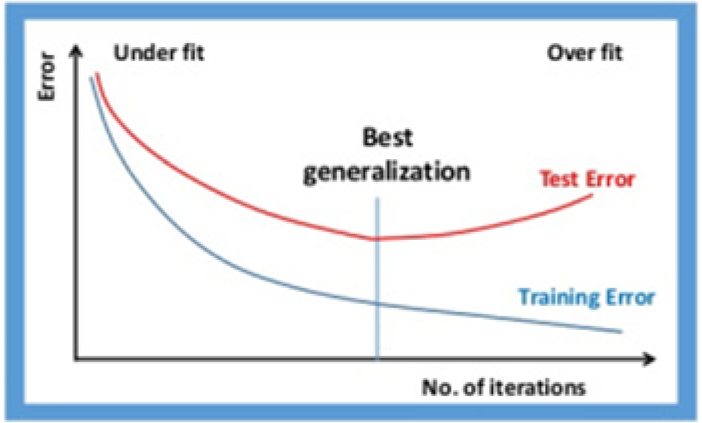
☑ ○ mnist_experiment/fit_model_6_batch_size=32,beta_1=0.24684,beta_2=0.80671,learning_rate=0.075683_2020-09-14_03-11-44v23ih_t9

☑ ○ mnist_experiment/fit_model_7_batch_size=64,beta_1=0.20124,beta_2=0.36737,learning_rate=0.034757_2020-09-14_03-12-22jfyg7ew7

	Name	Smoothed	Value	Step	Time	Relative
●	mnist_experiment/fit_model_0_batch_size=16,beta_1=0.12915,beta_2=0.49016,learning_rate=0.044842_2020-09-14_03-10-15c7wyntu6	34.17	43.58	4	Mon Sep 14, 03:11:06	31s
●	mnist_experiment/fit_model_1_batch_size=32,beta_1=0.53021,beta_2=0.32761,learning_rate=0.041112_2020-09-14_03-10-1909m6aqs3	21.29	26.95	4	Mon Sep 14, 03:10:50	18s
●	mnist_experiment/fit_model_3_batch_size=128,beta_1=0.40927,beta_2=0.64514,learning_rate=0.0095054_2020-09-14_03-11-17f7qhpj8c	10.58	12.96	4	Mon Sep 14, 03:11:36	7s
●	mnist_experiment/fit_model_4_batch_size=256,beta_1=0.13418,beta_2=0.3338,learning_rate=0.034093_2020-09-14_03-11-24o1twiowp	7.786	9.591	4	Mon Sep 14, 03:11:42	8s
●	mnist_experiment/fit_model_6_batch_size=32,beta_1=0.24684,beta_2=0.80671,learning_rate=0.075683_2020-09-14_03-11-44v23ih_t9	20.46	25.9	4	Mon Sep 14, 03:12:16	18s
●	mnist_experiment/fit_model_7_batch_size=64,beta_1=0.20124,beta_2=0.36737,learning_rate=0.034757_2020-09-14_03-12-22jfyg7ew7	14.31	18.7	4	Mon Sep 14, 03:12:47	13s

Understanding hyperparameters - Generalization

Generalization



- ANNs generalize well despite having many more parameters than samples.
- Test error vs Hidden units
- Especially interesting in distributed execution environments.
- Many hidden units within a layer with regularization techniques can increase accuracy.
- Optimal stopping time depends on the signal to noise ratio → Low quality data requires longer training time

Understanding hyperparameters - Bias-Variance tradeoff

- **A bias** in a model appears when the algorithm is not flexible enough to generalize well from the data.

  Eg. : Linear parametric algorithms with low complexity like Naive Bayes tend to have a high bias.

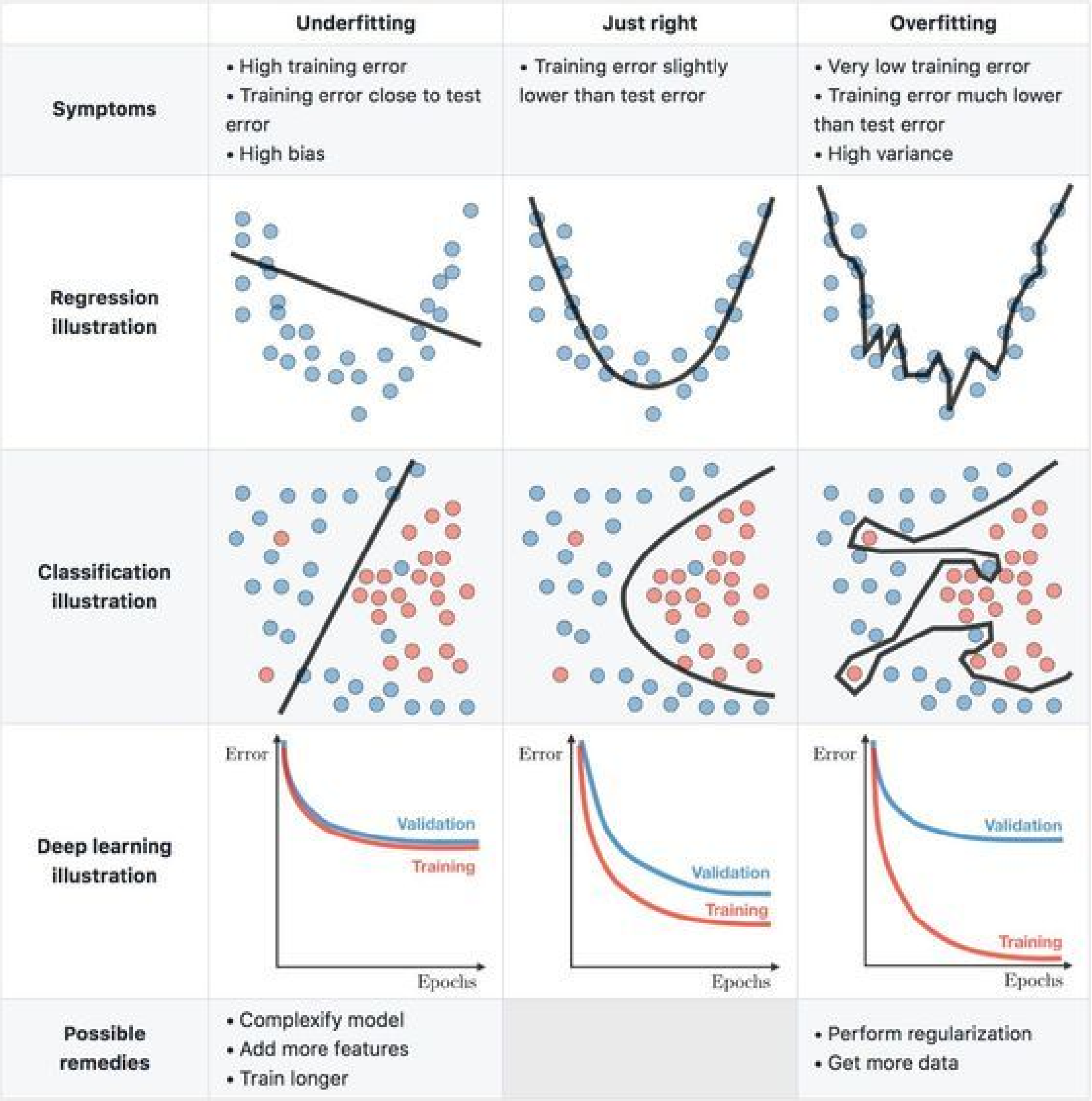
- **Variance** occurs in the model when the algorithm is sensitive and highly flexible to the training data.

  Eg.: Non-Linear non-parametric algorithms with high complexity such as Neural Networks tend to have high variance.

- There are various ways to find the balance of bias and variance for each algorithm family by using methods such as **regularization, pruning**, etc.

Understanding hyperparameters - Overfitting and Underfitting

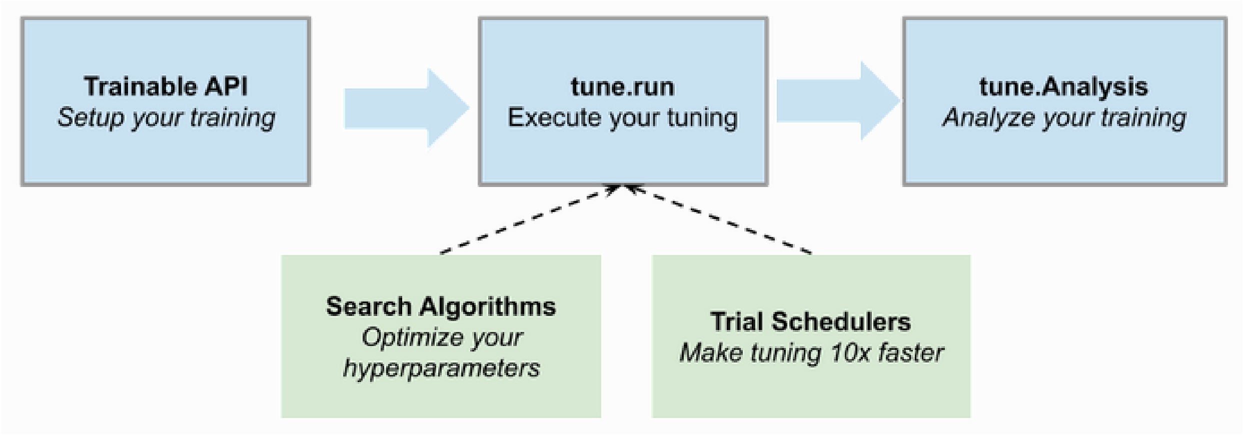
- Overfitting → bad generalization
- Smaller number of units may cause underfitting.
- Dropout is regularization technique to avoid overfitting (increase the validation accuracy) thus increasing the generalizing power. You are likely to get better performance when dropout is used on a larger network, giving the model more of an opportunity to learn independent representations.



	Underfitting	Just right	Overfitting
Symptoms	• High training error • Training error close to test error • High bias	• Training error slightly lower than test error	• Very low training error • Training error much lower than test error • High variance
Regression illustration			
Classification illustration			
Deep learning illustration	Error Validation Training Epochs	Error Validation Training Epochs	Error Validation Training Epochs
Possible remedies	• Complexify model • Add more features • Train longer		• Perform regularization • Get more data

CS 229 - Machine Learning Tips and Tricks Cheatsheet

Tuning parameters with automated tools – Ray Tune



Advanced trial schedulers

- ASHA
- Median Stopping Rule
- HyperBand
- BOHB
- Population Based Training

Many optimization algorithms

- Bayesian Optimization
- Bayesian Opt/HyperBand
- NevergradSearch
- Gradient-free Optimization
- SigOptSearch
- …

Learning rate

- The Learning Rate is a parameter of the Gradient Descent optimizer.
- Controls the change of weights for the network to the loss of gradient.
- Model training will progress very slowly for low learning rates as it would be making tiny adjustments to the weights in the network and it could also lead to overfitting.
- High learning rates tend to leaps bouncing around chaotically, not leading to the optimization of the objective function and missing the local optima making the training diverge. But having a large learning rate helps in regularizing the model well to capture the true signal.
- There are many choices of schedulers.
- The interaction between batch size and learning rate is very important for certain distribution schemes.
- Adaptive learning rate algorithms are prevalent nowadays.

Minibatch size

- Minibatch Size = number of training samples propagated through the network.
- Our goal to obtain the maximum performance by minimizing the computational time required.
- Batch size affects the training time.
- In general we want to maximize efficiency and utilization of memory.
- Rather modifying the batch size rather than changing the learning rate.
- Larger batchs enable using a large learning rate.
- Larger batch sizes tend to have low early losses while training.
- Final loss values are low when the batch size is reduced.

THANK YOU FOR YOUR ATTENTION

www.prace-ri.eu