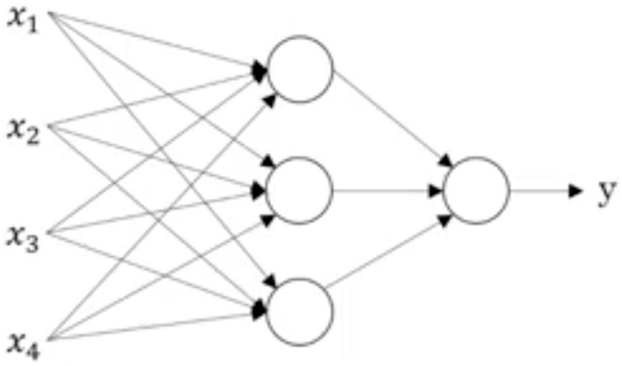
# PHY654

## Machine learning (ML) in particle physics

Swagata Mukherjee • IIT Kanpur

22nd August 2024

# Initialisation of w and b parameters




**Random initialisation** of weights (w) is necessary for NN.

Initializing w parameters to 0 does not work for NN. (all hidden units become same)

Use **small** random values. Large values may lead to vanishing gradient issue.

You can use **np.random.randn(i,j) * 0.01**

However, b can be initialized to 0.

# Vectorisation: why is it important?

Process of training a neural net is often **iterative** and **empirical** process.

Idea → Implement → Run and check performance → Repeat

This part should be fast

If your code is computationally non-optimal, then it may take a long time to run. That can become a huge bottleneck. That's why we will always try to use vectorization whenever possible.

Code to see how the speed-up works

https://github.com/swagata87/IITKanpurPhy654/blob/main/vectorisation.ipynb

# Parameters and hyperparameters

**Parameters: w and b**

Learned from the data during the training process

**Hyperparameters: learning rate, number of iterations, number of hidden layers, number of hidden units in each layer, choice of activation function.**

**Hyperparameters control / influence parameters.**

# Train / dev / test set


Recall that applied ML is an iterative process.
Idea → Code it up → Check performance → Repeat

Use training set to keep on training your NN.
Use dev set to check which network is performing best.
Once you are happy with a NN, check final performance in test set.

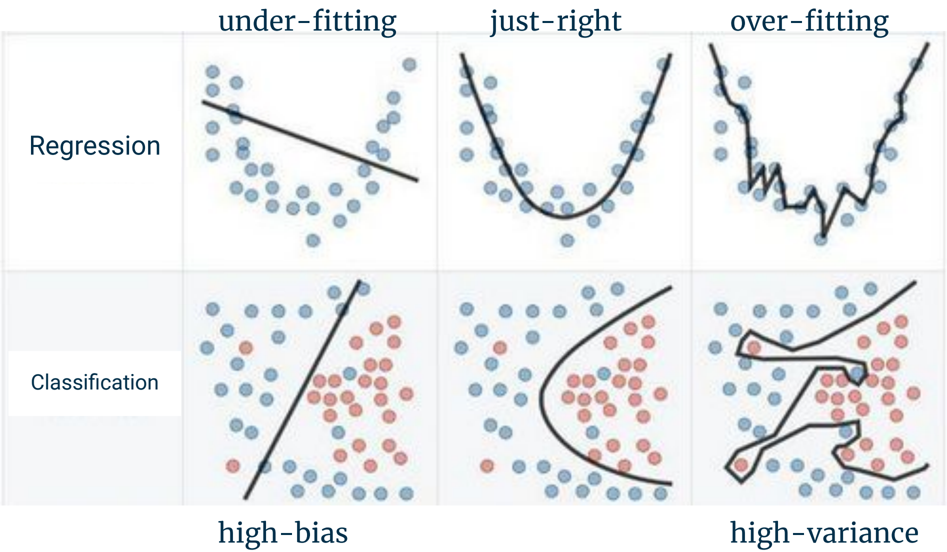
In modern era of big data, where we have **millions** of data,

**Train: 98%, Dev: 1%, Test: 1%**

If you have small amount of data (few **thousand**), then **60%, 20%, 20%** will be more appropriate.

In some cases, people do not have a test set, i.e, you can have only train and dev.

# Bias variance trade-off




1D or 2D example → so we can visualise this.

In high dimensional problems, no way to visualise this. Instead we have some metric that we can look at.

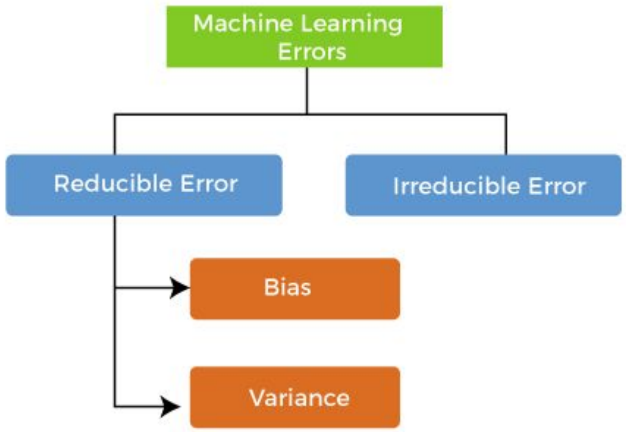
# Bias variance trade-off

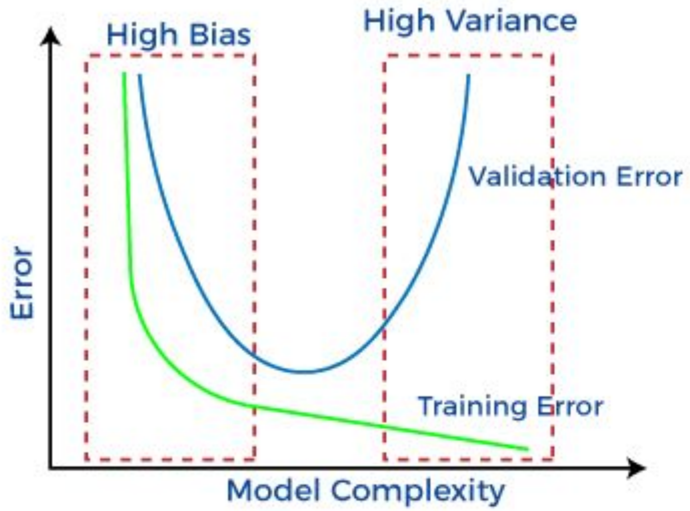
Two key numbers to look at to understand bias and variance are: Train Set error and Dev set error

Let's say you are writing a photon-vs-jet classifier

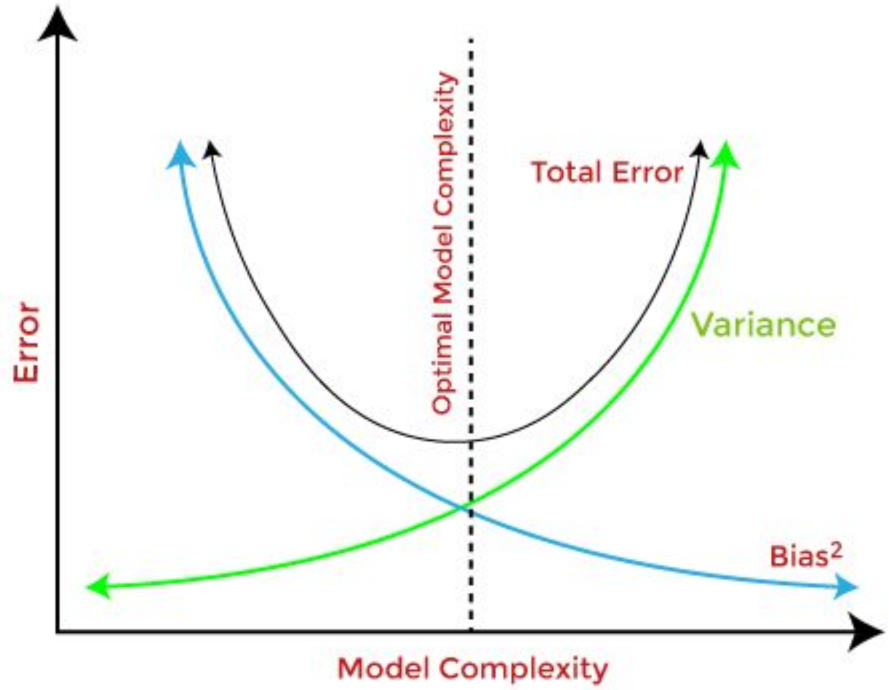
| | | | | |
|---|---|---|---|---|
| Train Set error | 1% | 15% | 15% | 0.5% |
| Dev Set error | 11% | 16% | 30% | 1% |
| | Doing well in training set. Doing poorly in dev set, ie, not generalising well. High Variance | Doing poorly in training set. Under-fitting. High bias. | High bias and high variance (this can also happen) | Low bias and low variance. Performance is okay. |

# Bias variance trade-off



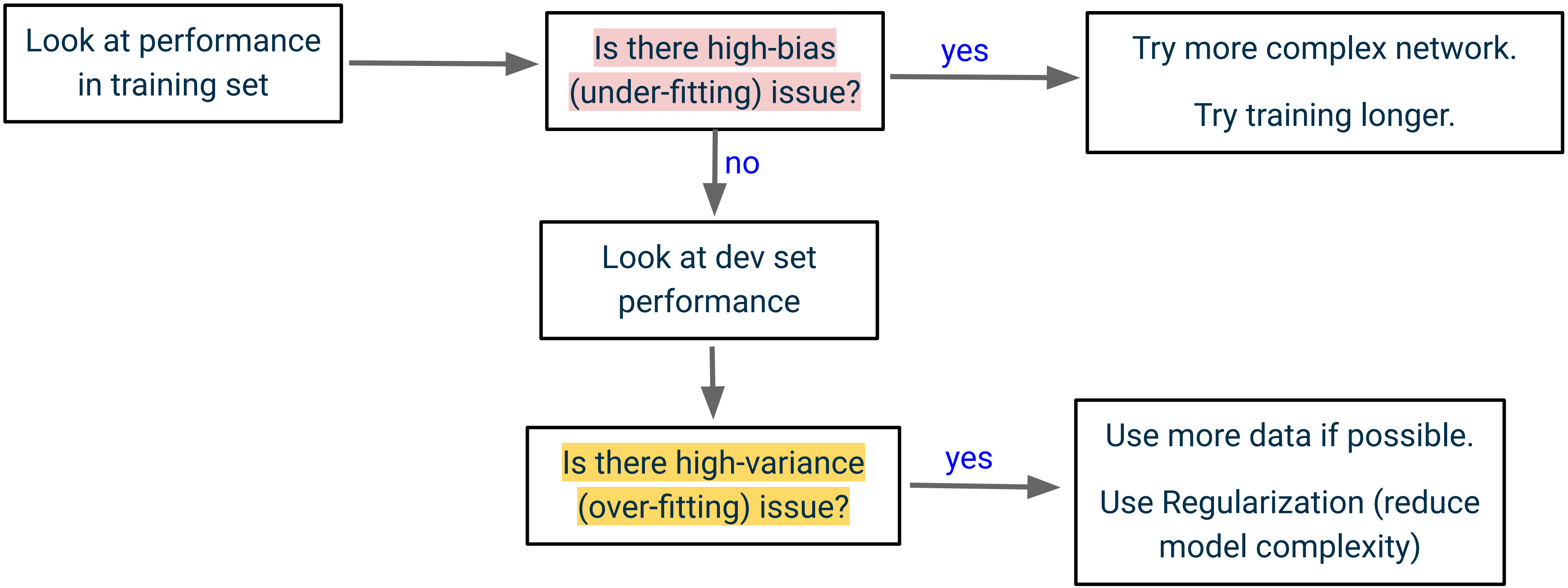
# Bias variance trade-off




Signature of high bias


Signature of high variance

# How to solve high-bias / high-variance?



# Next items

- Regularization
- Exponentially weighted average
- RMSprop
- Adam optimizer
- Mini-batch