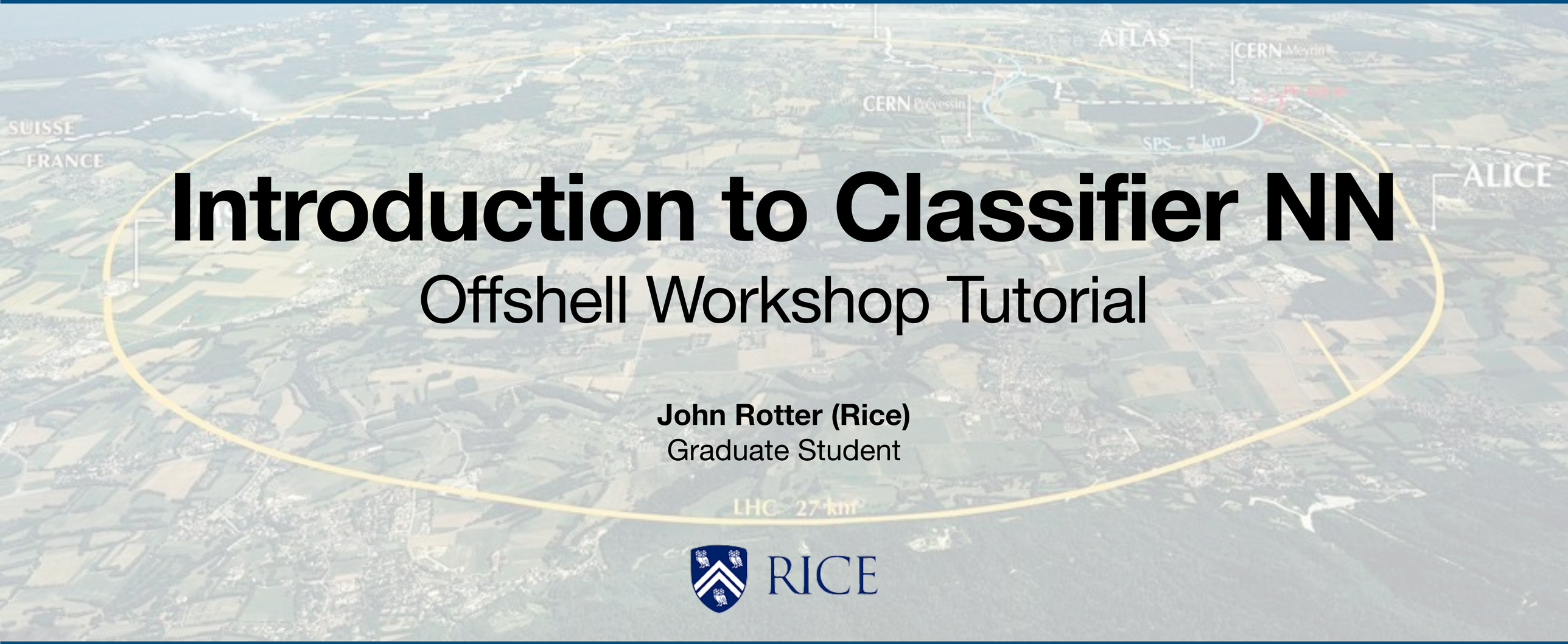


Introduction to Classifier NN

Offshell Workshop Tutorial

John Rotter (Rice)
Graduate Student

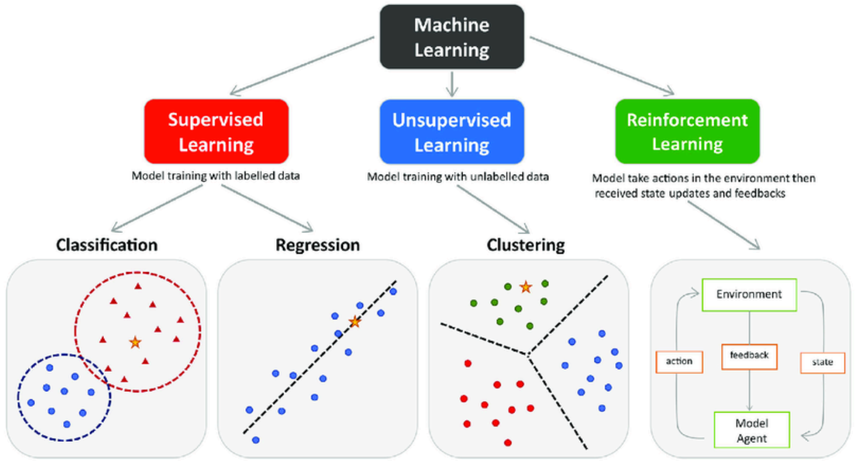
RICE

What is Machine Learning?

- Data driven general solution to a complex problem - Universal Function Approximators!

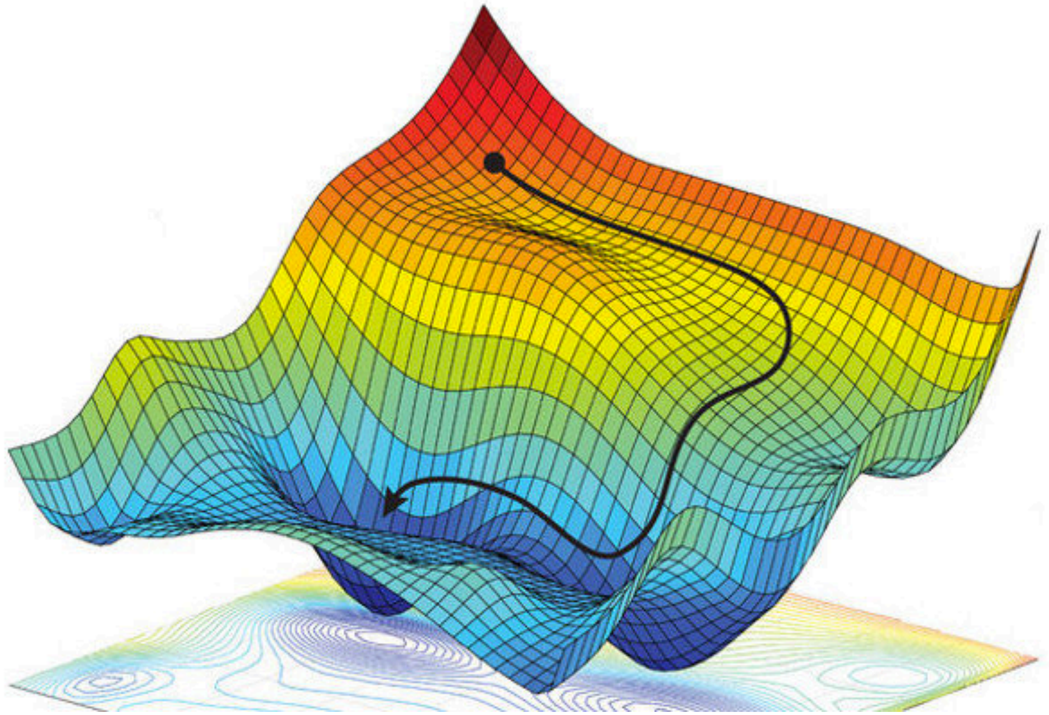
- The name “Learning” comes from personification of the algorithm but it is really just an optimization problem and **not a sentient being**

- Supervised Learning is when the algorithm is given a deterministic task
  - **Training involves a correction based on the prediction.**



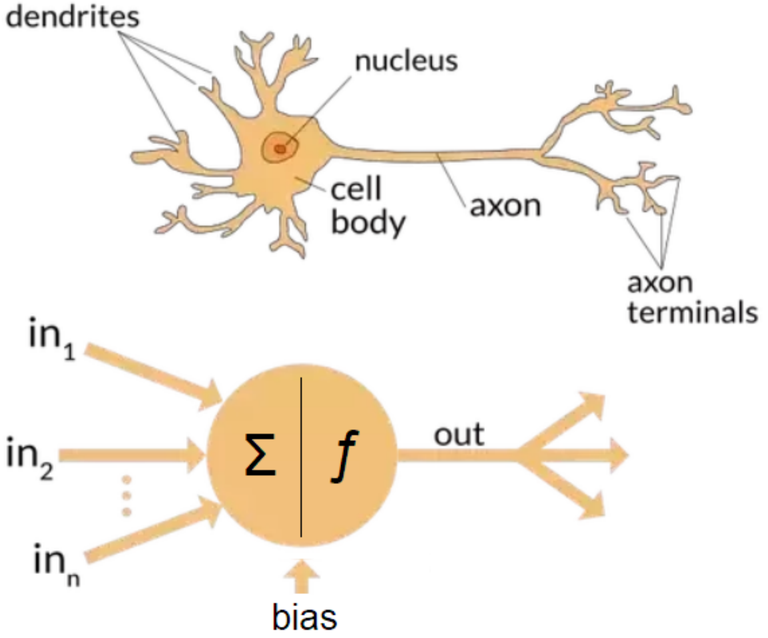
How to Reward the Machine

- To motivate an algorithm to perform as intended, its performance will need to be quantified
- Often this is done using a Loss Function which gives a penalty to wrong answer
- The Loss Function will return a larger value for wrong answers and zero for correct answers
- **Minimizing the Loss Function will lead to more and more correct answers**



Origins of Neural Networks

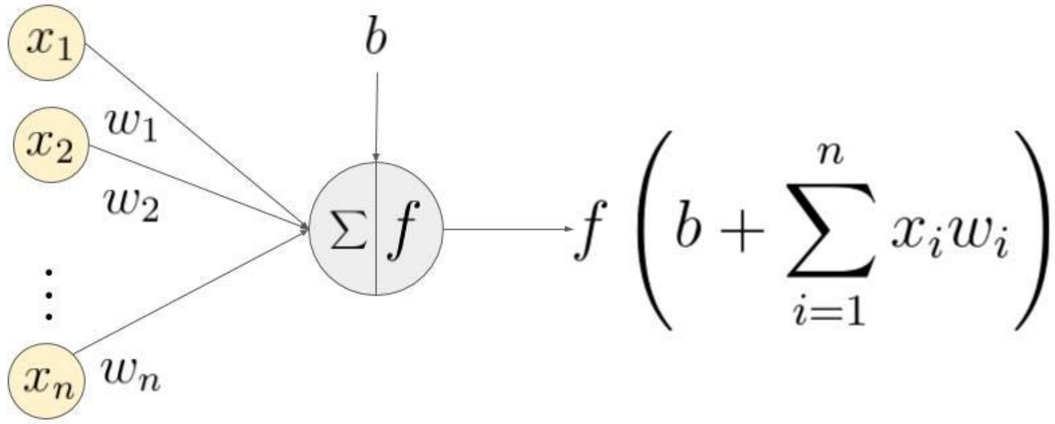
- Neural Networks were developed in an attempt to mimic how neurons in the brain function
- Neurons in the brain are wired to each other with each connection having its own weight
- Neurons also have their own internal behaviors and can send signals of their own to other neurons in the brain
- By building a simplified model of neurons and allowing them to wire together, we get a Neural Network



“Neurons the fire together, wire together”

The Neural Network Model

- Neural Networks (NN) have an input layer
- Each sequential layer is fully connected (dense) to the previous layer
- Each neuron's value is computed as:
  - Sum of all previous neuron's values, x_i, weighted by w_i
  - Plus an internal bias of the neuron, b
  - Then passed through an activation function, f
- This computation is done for every neuron in ever layer!
  - Since it is essentially matrix multiplication, this can be done very fast by the GPU!
- The final layer is the prediction of the network



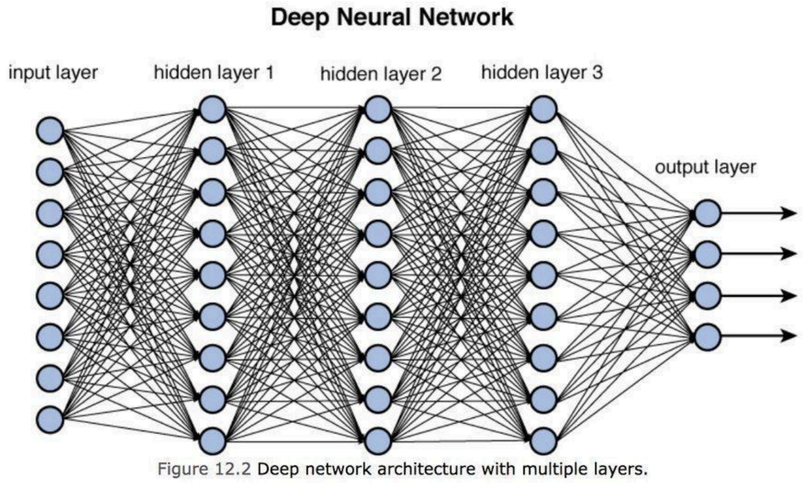
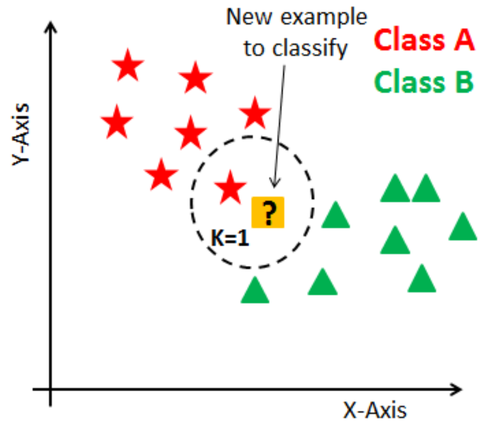


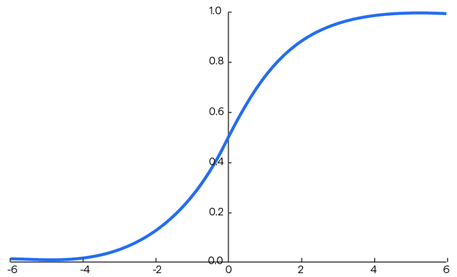
Figure 12.2 Deep network architecture with multiple layers.

Classifier NN

- Classifier NN are used to group datasets based on the features within the data
- Training of a Classifier NN is very similar to any other type of machine learning algorithm, however we need a way to **encode the different classes**
- The standard way to encode classes is to use one-hot encoding
  - Each class is represented as an orthogonal unit vector
- The output layer for a Classifier NN is often soft-maxed as to be interpreted as a probability of an event being in a given class



	BKG1	BKG2	SIG
N^1_{out}	1	0	0
N^2_{out}	0	1	0
N^3_{out}	0	0	1

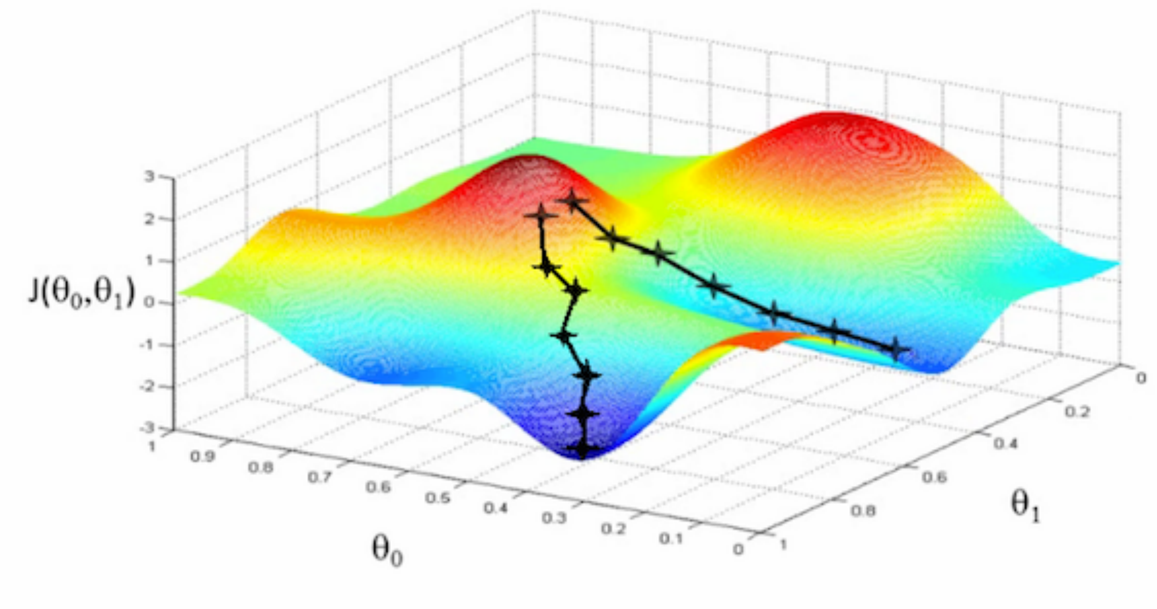


Training a Classifier NN

- When training any algorithm, a testing, training, and validation are randomly split to identify performance of the training
- Training data includes events that are already pre-classified and converted to one-hot encoding
- The input data is fed into the NN for each event and the output layer is soft-maxed
- The soft-maxed output and one-hot encoding are then compared in the loss function and assigned some loss
- Based on the loss function, the NN will update its sets of weights and biases to minimize the loss

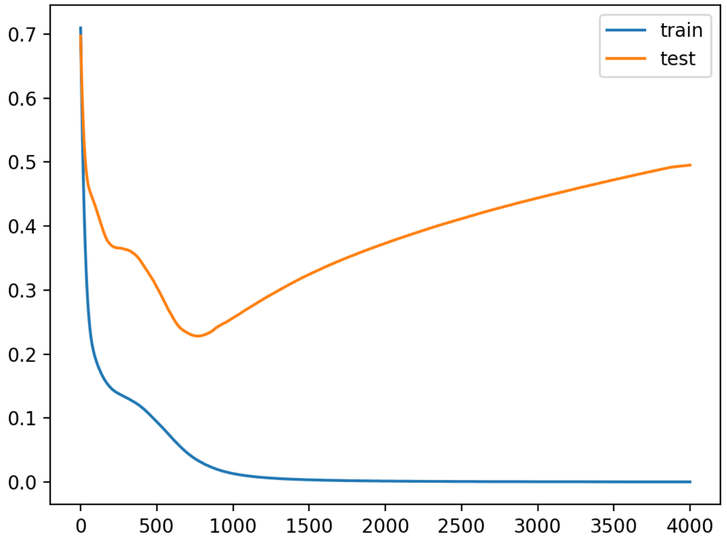
Batches, Stumbling, and Weighting

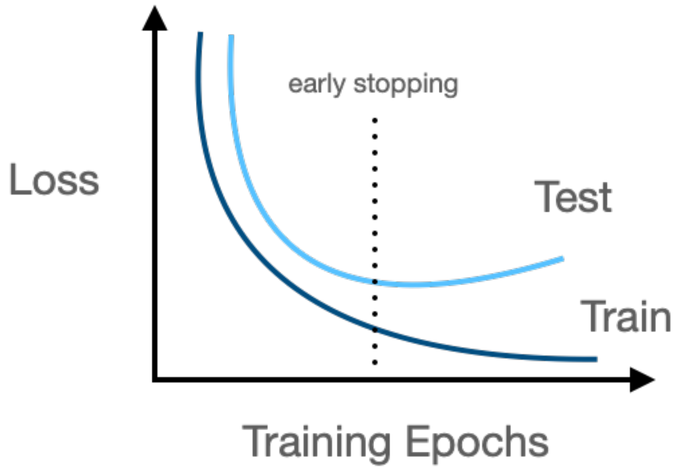
- Batch size is the number of training events before updating the algorithm
- The algorithm is then updated based on the gradient of the loss function and the algorithm “stumbles” forward
- Training events can be weighted differently in the loss function based on their significance to the training



Overtraining

- **Overtraining** is one of the major problems that can go wrong with a NN! It is identified by a difference in loss or accuracy between training and validation sets
  - Can be caused by having too many unconstrained variables in the training (i.e. 4 input variables in a NN with 100 million connections)
  - Can also be caused by training too many iterations of the NN for the training dataset's size (i.e. training for 15,000 epochs on a dataset of 2000 events)
  - What is actually happening is the training set is being learned itself instead of the relationship between variables - Effectively the NN is memorizing the training set





Hyperparameters

- Classifier NN have many hyperparameters that can be changed to produce different results in the training

- To name a few:
  - Batch Size
  - Number of times the NN updates (Number of Epochs)
  - Activation function of each layer (ReLu, Leaky ReLu, Tanh, etc)
  - Number of nodes in each layer
  - Early stopping (size of validation sets, patience, thresholds)
  - Event weights
  - Loss function
  - Optimizer Algorithm
  - Layer types (Dense, Sparse, Quantized)

The Tutorial

- We will be training a classifier NN to learn to identify if an event is in a background or signal category.
- We will be looking at simulated events that contain mass, p_T, and η values which might be seen in an analysis setting
- We will also plot many metrics of the training and discuss potential changes to the code!

https://github.com/Offshell-Workshop-LPC/Offshell-Workshop-ML-Tutorials

The Tutorial - Datasets