## Classification using Neural Network: Basketball vs Football Players

Luc Rulinda

12/6/2021

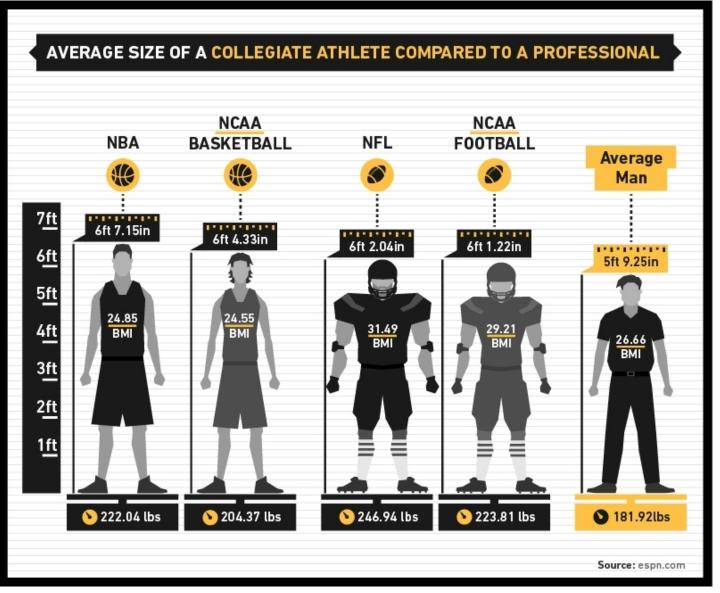
### Task at hand:

Designing a neural network that classifies various points in two categories using M- dimensional feature vectors.

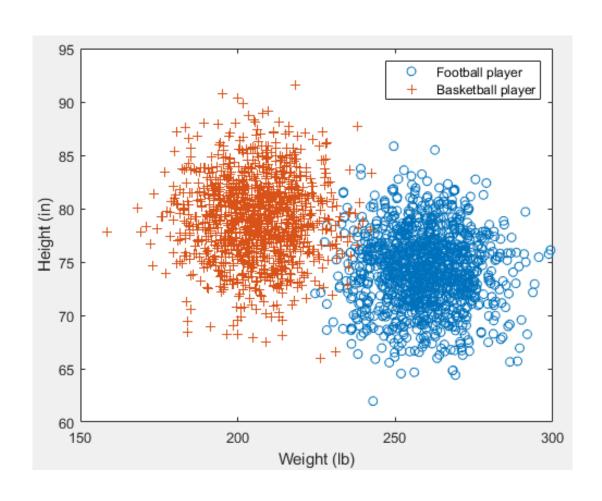
## Data used

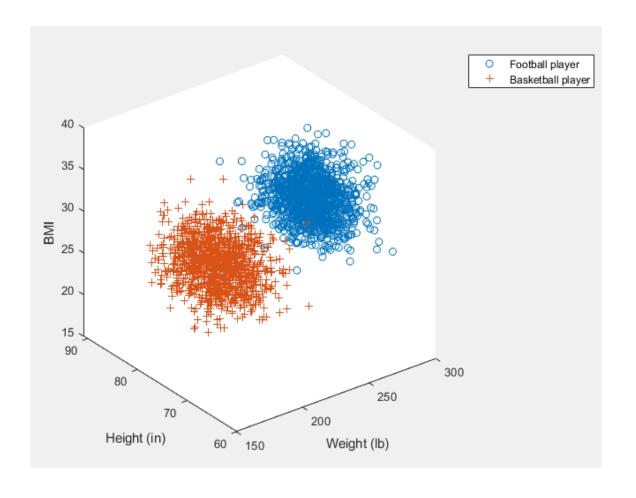
- 2 categories: Basketball players and Football players
- Feature vectors: height, weight and BMI (Ref: Sports Encyclopedia of Pro Football and Official NBA basketball Encyclopedia)
- Normal distribution of points

		Football Player	Basketball Player
	Mean (μ)	259.6	205.8
Weight (lbs)	S.D(σ)	12.1	12.9
	Mean (μ)	74.14	79
Height (in)	S.D(σ)	3.51	3.89



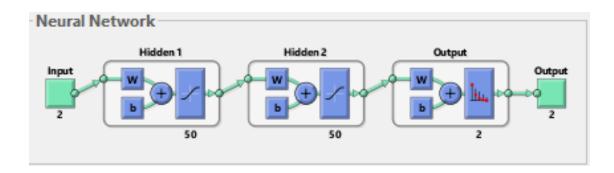
## M-dimensional feature vectors (2D vs 3D)





## Training the neural network

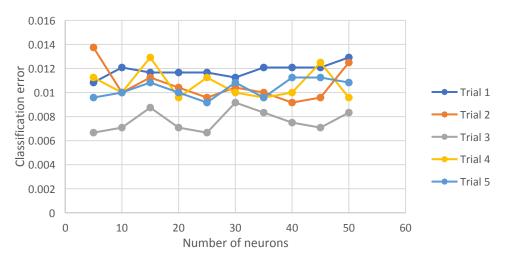
- "Patternet" function in Matlab
- Returns a pattern recognition networks (feedforward network)
   that can be trained to classify inputs according to target classes.
- Training function: "traingdx"- Variable Learn. Gradient descent
- Performance function: "crossentropy"



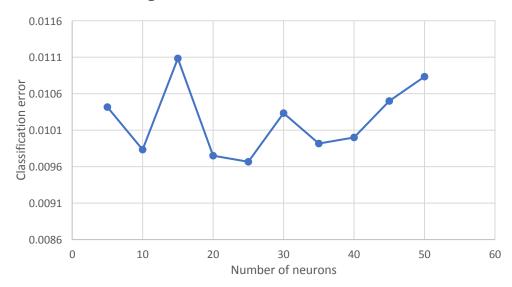
# Experimenting with various number of neurons:

Hidden 1 Hidden 2 Output Algorithms Data Division: Random (dividerand) Training: Gradient Descent with Momentum and Adaptive LR (traingdx) Performance: Cross Entropy (crossentropy) Calculations: MEX Progress 102 iterations 1000 Epoch: 0:00:01 Time: 0.601 0.0196 1.00e-05 Performance: 2.76 0.00588 1.00e-05 Gradient: Validation Checks: Hidden 1 Hidden 2 Output Output 35 35 40 40 45 45 2 50

#### Classification error vs number of neurons

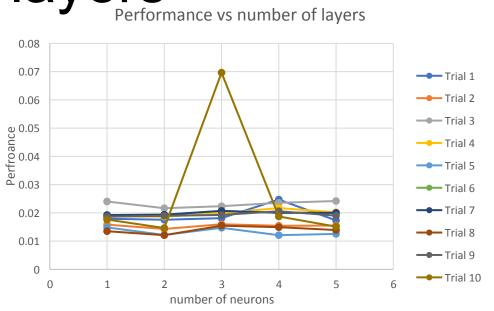


#### Average error vs number of neurons

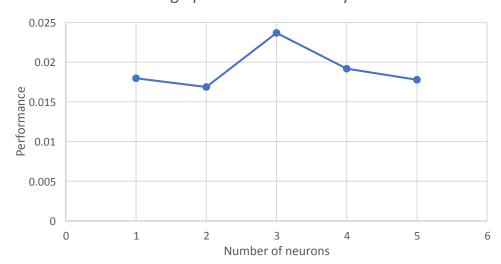


Experimenting with multiple hidden layers

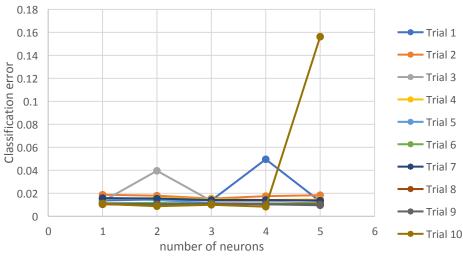
Classification error vs number



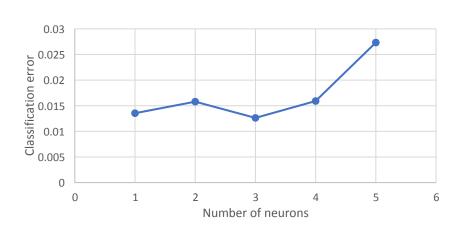
Average performance vs # layers



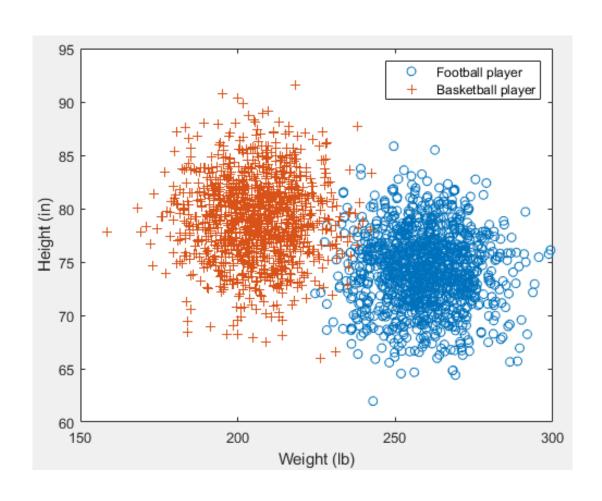
#### Classification error vs number of layers

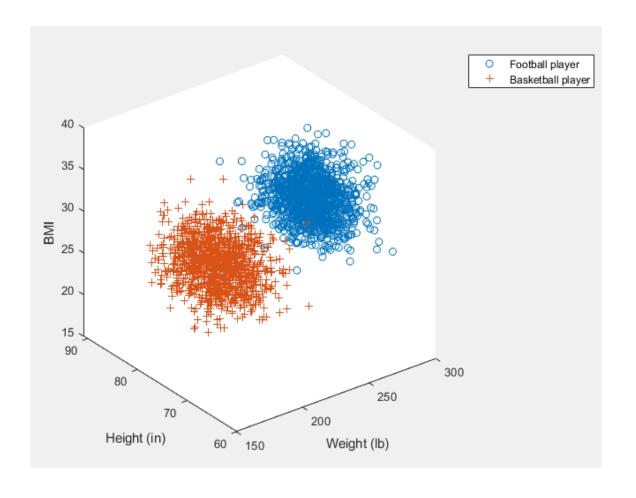


Average Classification error vs number of layers

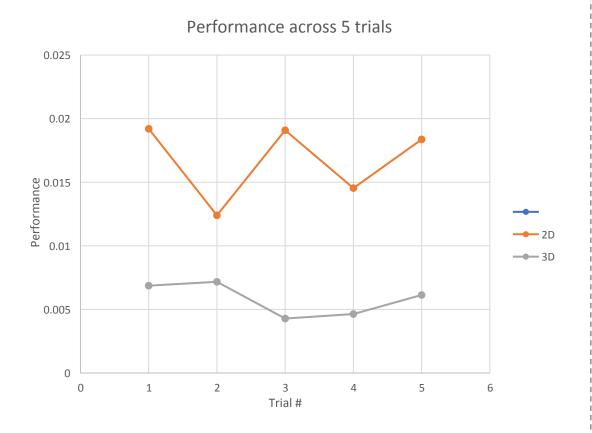


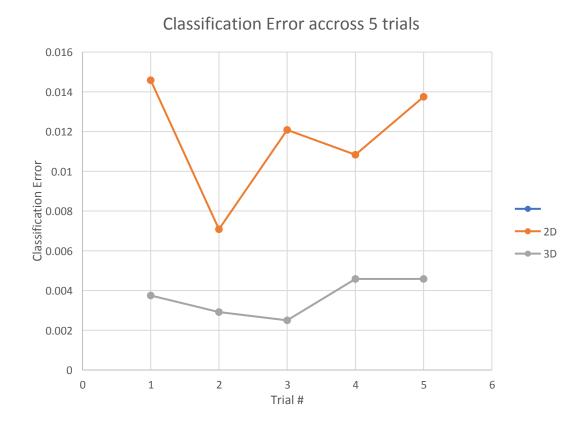
## M-dimensional feature vectors (2D vs 3D)





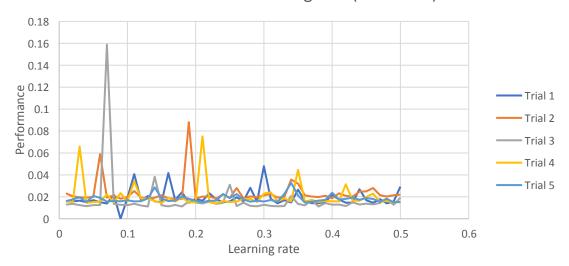
Performance and Accuracy (2 D vs 3D)



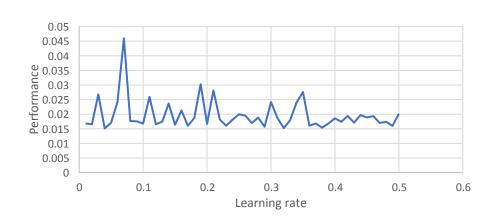


## Learning rate

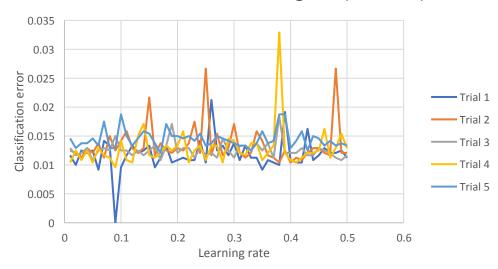
Performance vs Learning rate (0.01 - 0.5)



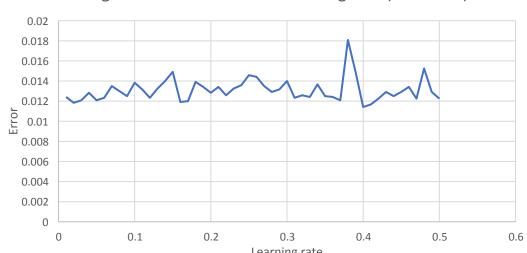
#### Average performance vs Learning rate



#### Classification error vs Learning rate (0.01 - 0.5)



#### Average classification error vs learning rate (0.01 - 0.5)



Thank you for your time!