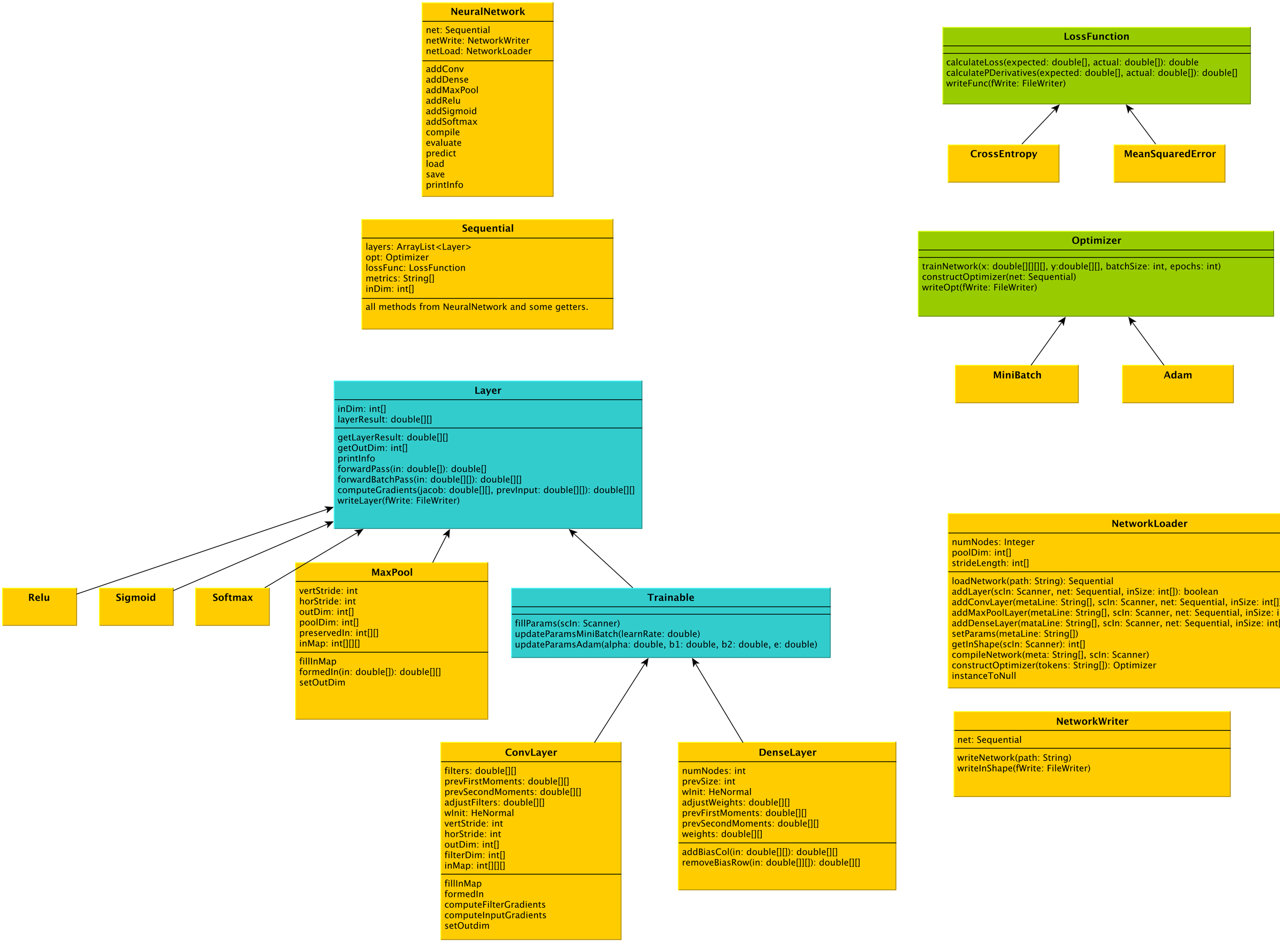
NeuralNetwork
net: Sequential
netWrite: NetworkWriter
netLoad: NetworkLoader
addConv
addDense
addMaxPool
addRelu
addSigmoid
addSoftmax
compile
evaluate
predict
load
save
printInfo
Sequential
layers: ArrayList<Layer>
opt: Optimizer
lossFunc: LossFunction
metrics: String[]
inDim: int[]
all methods from NeuralNetwork and some getters.
Layer
inDim: int[]
layerResult: double[][]
getLayerResult: double[][]
getOutDim: int[]
printInfo
forwardPass(in: double[]): double[]
forwardBatchPass(in: double[][]): double[][]
computeGradients(jacob: double[][], prevInput: double[][]): double[][]
writeLayer(fWrite: FileWriter)
Relu
Sigmoid
Softmax
MaxPool
vertStride: int
horStride: int
outDim: int[]
poolDim: int[]
preservedIn: int[][]
inMap: int[][][]
fillInMap
formedIn(in: double[]): double[][]
setOutDim
Trainable
fillParams(scIn: Scanner)
updateParamsMiniBatch(learnRate: double)
updateParamsAdam(alpha: double, b1: double, b2: double, e: double)
ConvLayer
filters: double[][]
prevFirstMoments: double[][]
prevSecondMoments: double[][]
adjustFilters: double[][]
wInit: HeNormal
vertStride: int
horStride: int
outDim: int[]
filterDim: int[]
inMap: int[][][]
fillInMap
formedIn
computeFilterGradients
computeInputGradients
setOutdim
DenseLayer
numNodes: int
prevSize: int
wInit: HeNormal
adjustWeights: double[][]
prevFirstMoments: double[][]
prevSecondMoments: double[][]
weights: double[][]
addBiasCol(in: double[][]): double[][]
removeBiasRow(in: double[][]): double[][]
LossFunction
calculateLoss(expected: double[], actual: double[]): double
calculatePDerivatives(expected: double[], actual: double[]): double[]
writeFunc(fWrite: FileWriter)
CrossEntropy
MeanSquaredError
Optimizer
trainNetwork(x: double[][][][], y:double[][], batchSize: int, epochs: int)
constructOptimizer(net: Sequential)
writeOpt(fWrite: FileWriter)
MiniBatch
Adam
NetworkLoader
numNodes: Integer
poolDim: int[]
strideLength: int[]
loadNetwork(path: String): Sequential
addLayer(scIn: Scanner, net: Sequential, inSize: int[]): boolean
addConvLayer(metaLine: String[], scIn: Scanner, net: Sequential, inSize: int[]): boolean
addMaxPoolLayer(metaLine: String[], scIn: Scanner, net: Sequential, inSize: int[]): boolean
addDenseLayer(mataLine: String[], scIn: Scanner, net: Sequential, inSize: int[]): boolean
setParams(metaLine: String[])
getInShape(scIn: Scanner): int[]
compileNetwork(meta: String[], scIn: Scanner)
constructOptimizer(tokens: String[]): Optimizer
instanceToNull
NetworkWriter
net: Sequential
writeNetwork(path: String)
writeInShape(fWrite: FileWriter)