Notes for Final

6/22/2024

SVM

* Support Vector Machines are best suited for classification, small to medium data. – text p 175
* Soft Margin Classification less sensitive to outliers than hard. P. 176
* Linear SVM requires data that is linearly separable.
* Can use polynomial features on linear SVM in scikit for nonlinear SVM. P.179
* SVM regression is possible p 184
* K Means is non neural net approach to classifying unlabeled data.

Decision Trees

* Can handle both regression and Classification p. 195
* Split nodes ask more questions, leaf nodes are answer
* Gini impurity = node is pure (gini=0) if all training instances it applies to belong to the same class p. 197
* CART algorithm only has 2 children per split node
* Decision Trees are white box algorithms. Random forests and neural nets are black box p. 199
* CART is greedy; searches for optimum split at the top level.
* Complexity of trees is O(log(m)/log(2)) since they are mostly balanced.
* Entropy is another impurity measure, entropy = 0 when all belong to one class. P. 201
* Regression trees try to minimize MSE instead of impurity.
* Data shape very important for trees p. 206
* Main negative is high variance.

Neural Nets

* Logical neurons diagram p. 303. Binary inputs and one binary output.
* Perceptron is TLU, and inputs and outputs are numbers p 304. Perceptron is a layer of TLU’s all connected to every input, this is a dense layer (fully connected layer).
* How to train perceptron: for every output neuron that produced a wrong prediction, it reinforces the weight that would have led to right prediction. P. 306
* Backpropagation is combo of reverse-mode autodiff and gradient descent p. 310
* Handles mini batches and fully trains data multiple times, epochs.
* Mini batch passes forward through all layers, next the algo measures the output error using loss function.
* Then it calculates how much each connection to the output layer contributed to error using chain rules. The algo then moves back down each layer calculating the same by propagating the error gradient backwards.
* Finally, algo performs gradient descent step to tweak connections p 311
* Good activation functions: sigmoid, Relu, others p. 312
* P. 316 modern MLP diagram
* Keras sequential model is single stack of layers connected sequentially p. 319
* Loss functions improve gradient descent, not autodiff p. 324
* Fine tune models by changing number of hidden layers, number of neurons per hidden layer, learning rate & batch size, optimizer and activation functions p. 349
* Activation function transforms input into output, the loss function calculates how well the neuron performs.
* **Activation Function**: This function is applied to the output of a neuron or layer of neurons, transforming the input signal into an output signal. It decides whether a neuron should be activated or not, based on whether each neuron's input is relevant for the model's prediction. Activation functions introduce non-linearity into the network, enabling it to learn complex patterns. Common examples include ReLU (Rectified Linear Unit), Sigmoid, and Tanh functions.
* **Loss Function**: Also known as the cost function, it measures how well the neural network performs by comparing the predicted outputs of the network to the actual target values. The loss function quantifies the difference between the two, guiding the optimization process during training. The goal of training is to minimize this loss, improving the model's accuracy. Examples of loss functions include Mean Squared Error (for regression tasks) and Cross-Entropy Loss (for classification tasks).