INFO251 – Applied Machine Learning

Lab 8

Emily Aiken

Announcements

- PS4 solutions posted tomorrow
- PS5 due March 30

Remaining Labs and Assignments

- **Lab 8**: Gradient boosting and basic neural networks
- **Lab 9**: Convolutional and recurrent neural networks
  - Problem set 5: Trees, forests, and networks, due March 30
- **Lab 10**: Fairness in machine learning
  - Problem set 6: Fairness in machine learning, due April 11
- **Lab 12**: Unsupervised learning
  - Problem set 7: Unsupervised and deep learning, due April 25
- **Lab 13**: TBD

Gradient Boosting

- **Bagging**: Build an ensemble of "weak learners" where each tree is trained on a random subset of the data
- **Boosting**: Build learners sequentially; data points that are misclassified in the model with *k* trees are upweighted when training the *k+1* tree
  - Adaboost: Fit the *k+1* model on the residual from the k^{th} model
- How does boosting relate to gradient descent?
  - Residuals are negative gradients
  - Updating F based on the residual = updating F in the direction of the negative gradient

$$L(y, F(x)) = (y - F(x))^2/2$$

$$\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} = F(x_i) - y_i$$

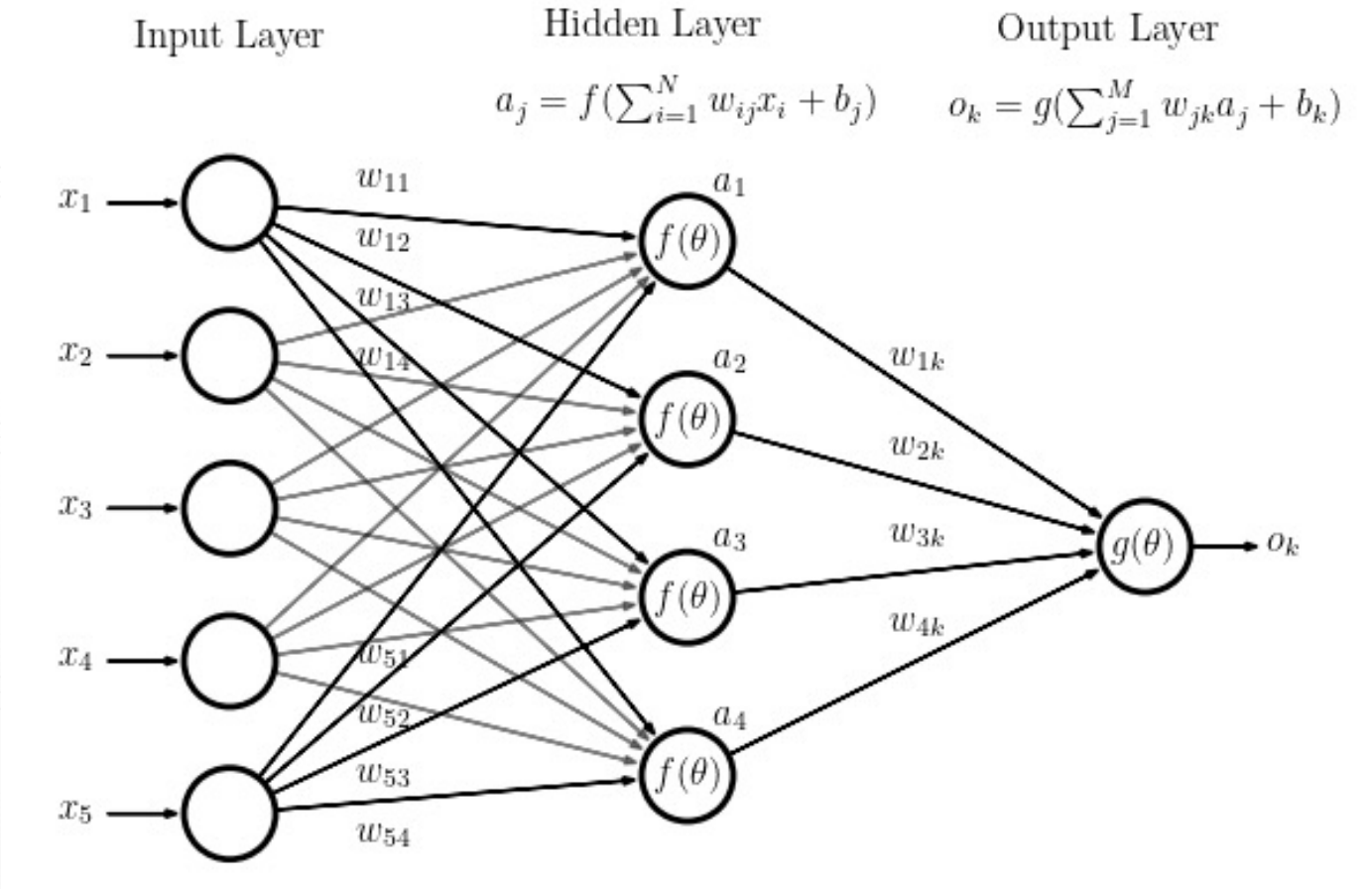
Gradient Boosting Implementations

dmlc
XGBoost

Yandex
CatBoost

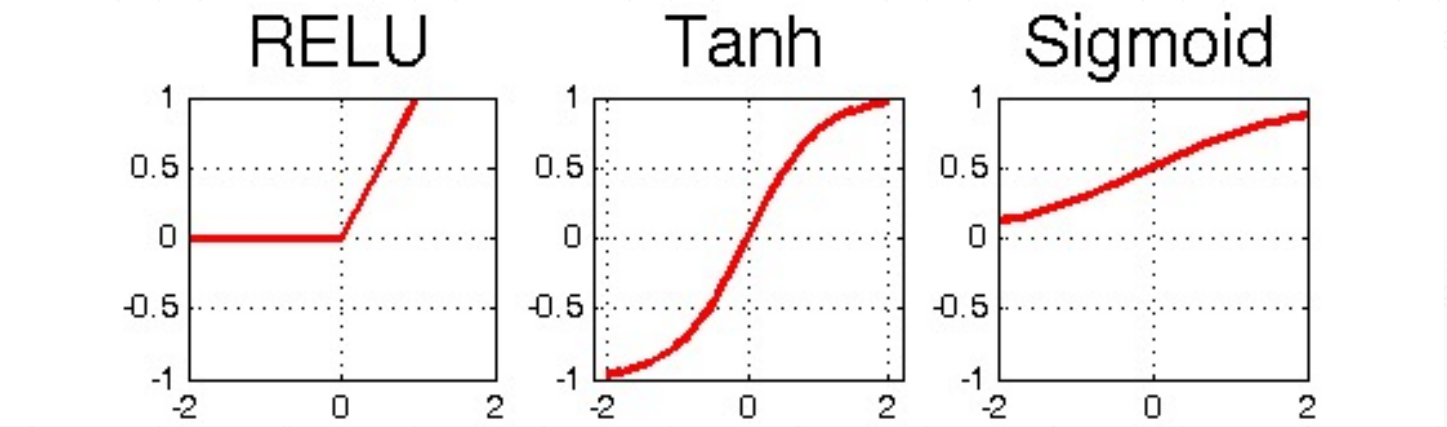
LightGBM

Neural Networks



Activation Functions

- Output layer activation
  - Same as the loss function for standard supervised models
  - Regression: No activation necessary
  - Classification: Sigmoid
- Hidden layer activations
  - Sigmoid
  - Relu (hinge)
  - Hyperbolic tangent



Optimization

- Loss functions
  - Regression: Mean squared error, mean absolute error
  - Binary Classification: Logistic (binary cross-entropy)
  - Multiclass Classification: Cross-entropy
- Optimization algorithm: Backpropogation
  - Implementations: Stochastic gradient descent, Adagrad, Adam, etc.

Hyperparameters

- Network structure
  - Number of hidden layers
  - Number of nodes in each hidden layer
  - Hidden layer activation function
- Optimization
  - Optimizer
  - Learning rate
  - Batch size
  - Number of iterations (epochs), early stopping

Neural Network Implementations

TensorFlow

PyTorch