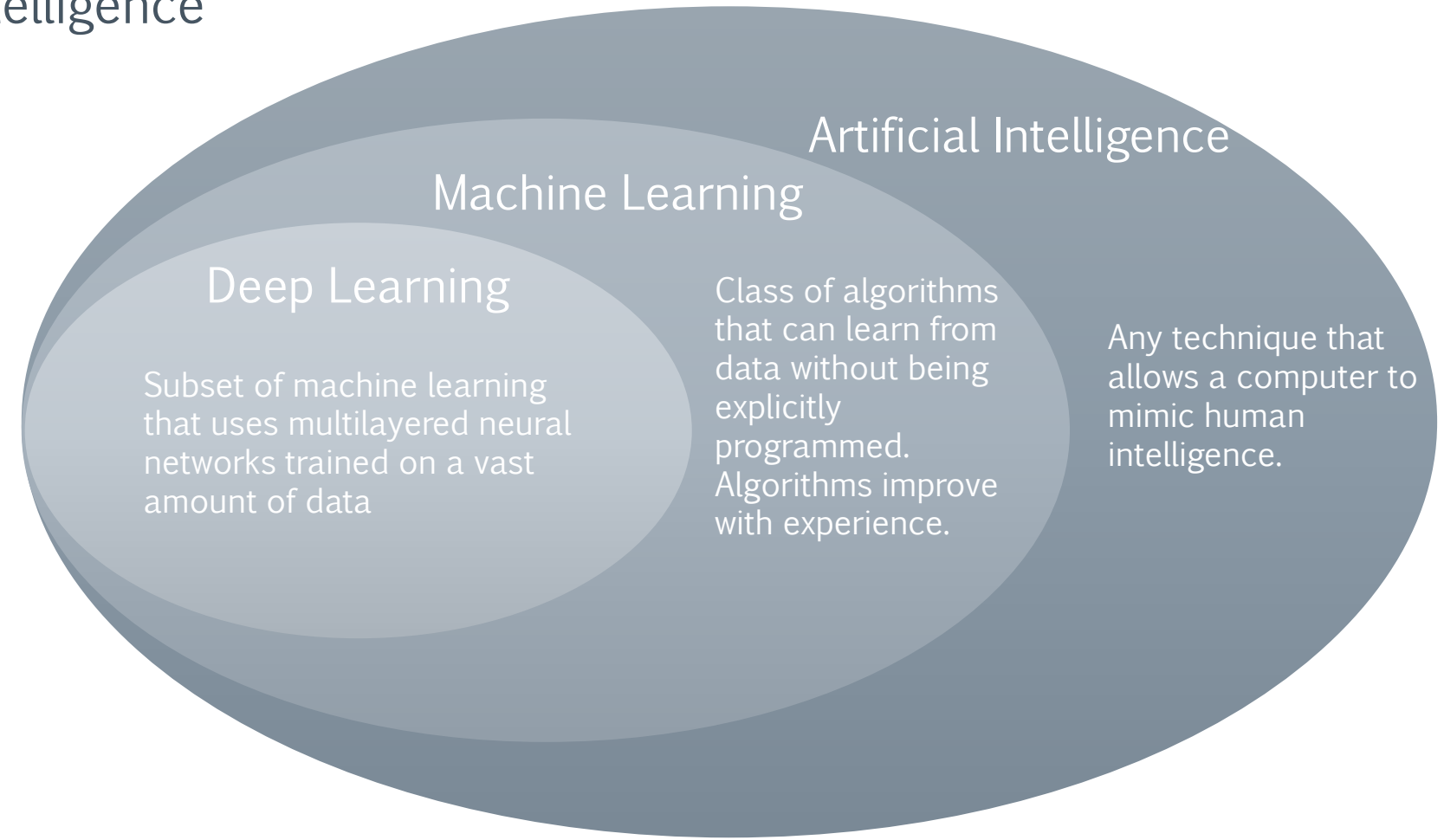
Artificial Intelligence in Medical Imaging

Colin Schaeffer

AI Hierarchy

- Goal of AI: Create machines that posses same characteristics as human intelligence



- Narrow AI: Programmed to perform a single task

Classifications of Machine Learning Algorithms

Three classes of Machine Learning algorithms:

1. Supervised Learning
   - Requires labeled training data
   - Direct feedback

2. Unsupervised Learning
   - Unlabeled training data
   - Finds hidden structure in data

3. Reinforcement Learning
   - Models decision process
   - Reward system

Learning Objectives

1. Understand the inner workings of an Artificial Neural Network

2. Understand the inner workings of a Convolutional Neural Network

3. Understand how these algorithms can be applied to the field of medical physics

Outline

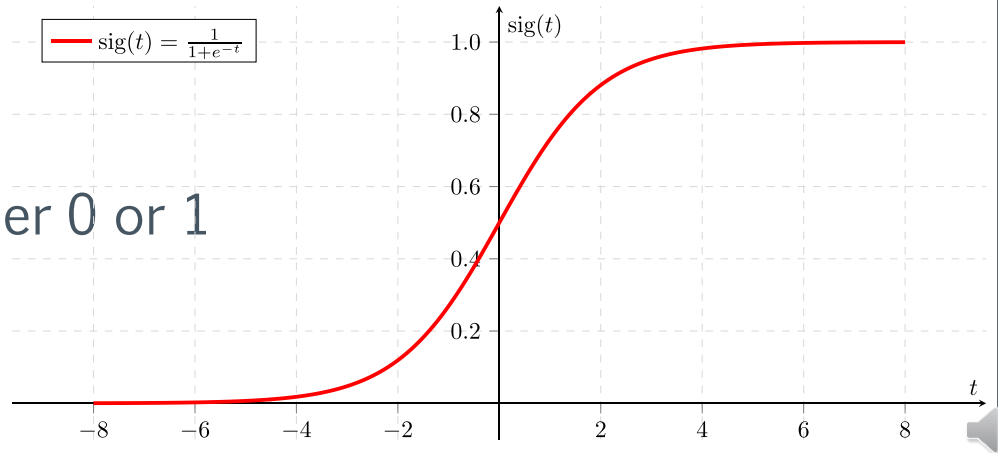
1. Logistic Regression
2. Artificial Neural Network
3. Convolutional Neural Network
4. Applications in Medical Imaging

Logistic Regression: Model

- Binary Classifier: Output $= \hat{y} \in \{0,1\}$
  - Sigmoid function: $Sig(x) = \frac{1}{1+e^{-x}}$

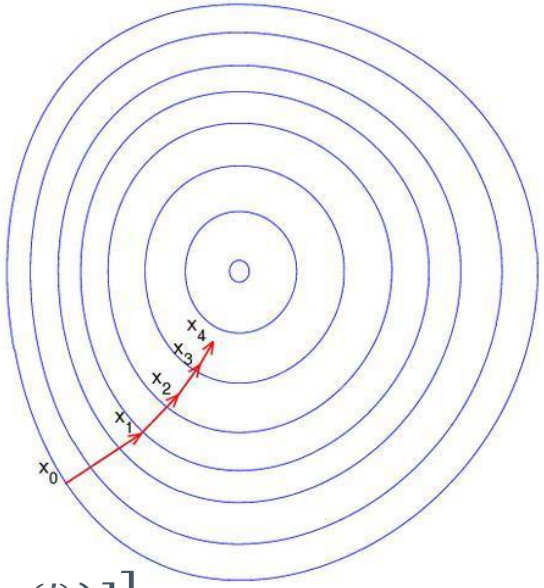
The model is defined as thus:

1. Model takes a set of parameters $x = \{x_0, x_1, x_2, \dots\}$ as input
   - Parameters chosen by self ($x_0 = 1$ by default)
2. Each parameter is assigned a weight W $= \{w_0, w_1, w_2, \dots\}$
   - Initial weights are chosen randomly
3. Model outputs a value $0 \leq$ Output ≤ 1
   - Net input = $w_m x_m + \cdots w_1 x_1 + w_0$
   - Output given by $Sig(W^T x) = \frac{1}{1+e^{-\sum_i w_i x_i}}$
4. Threshold is used to classify output to either 0 or 1
   - Ex: With threshold value $0 \leq T \leq 1$:
     - If $Sig(W^T x) < T$ then case is classified as 0
     - If $Sig(W^T x) \geq T$ then case is classified as 1



Logistic Regression: Training

1. Use N training cases where each case has their own input vector X and label y = {0,1}

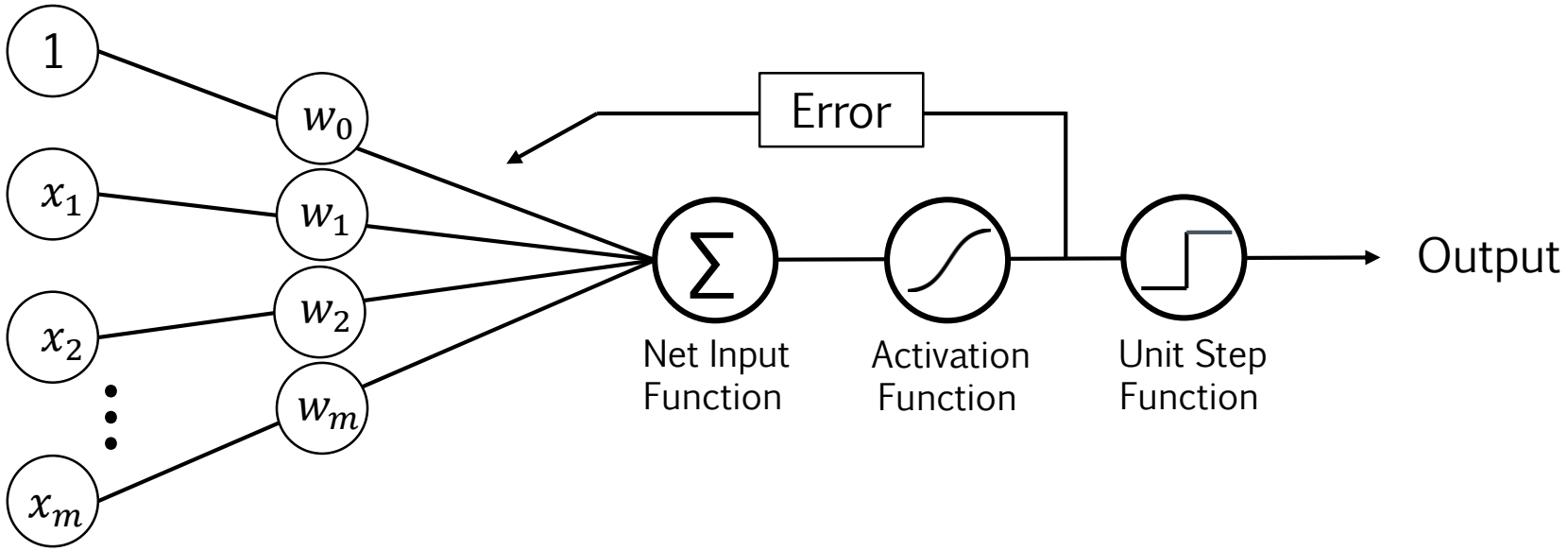


2. Run each case through model

3. Apply cost function to model output
   - Cost Function: Binary Cross Entropy
   - $E(W) = -\frac{1}{N}\left[\sum_i^N y^{(i)} \log\left[Sig_W\left(x^{(i)}\right)\right] + \left(1 - y^{(i)}\right) log\left[1 - Sig_W\left(x^{(i)}\right)\right]\right]$

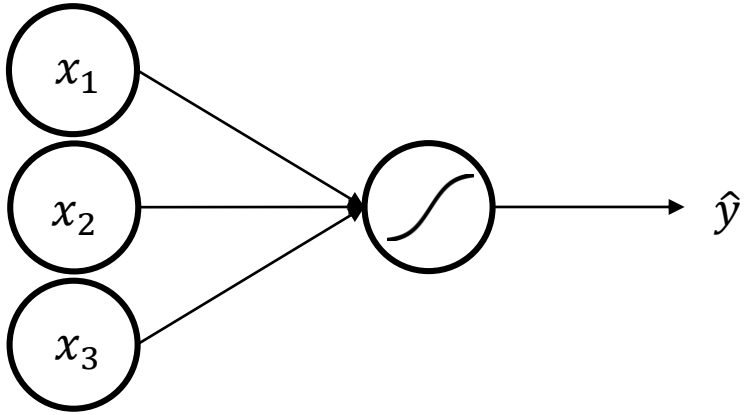
4. Update weights by gradient descent
   - $W = W - \alpha \nabla E$
   - Where α is the learning rate

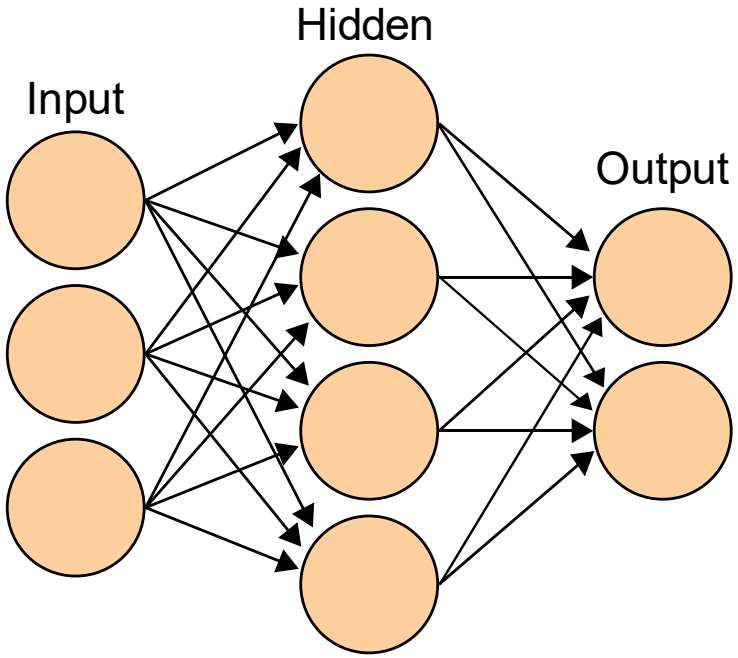
Logistic Regression: Overview



Artificial Neural Networks (ANN)

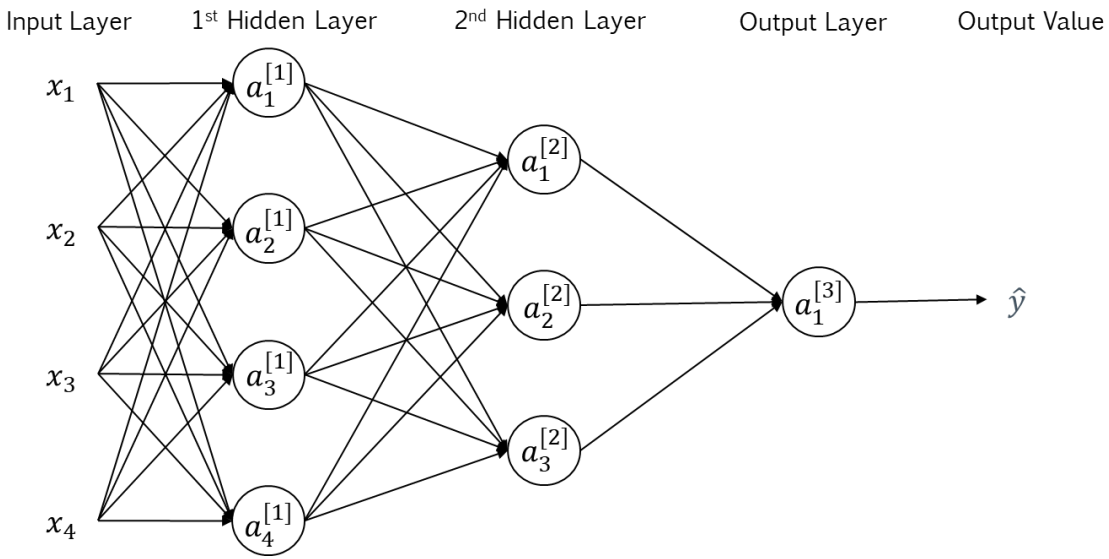
› Fully connected Neural Network is just a collection of logistic regressions



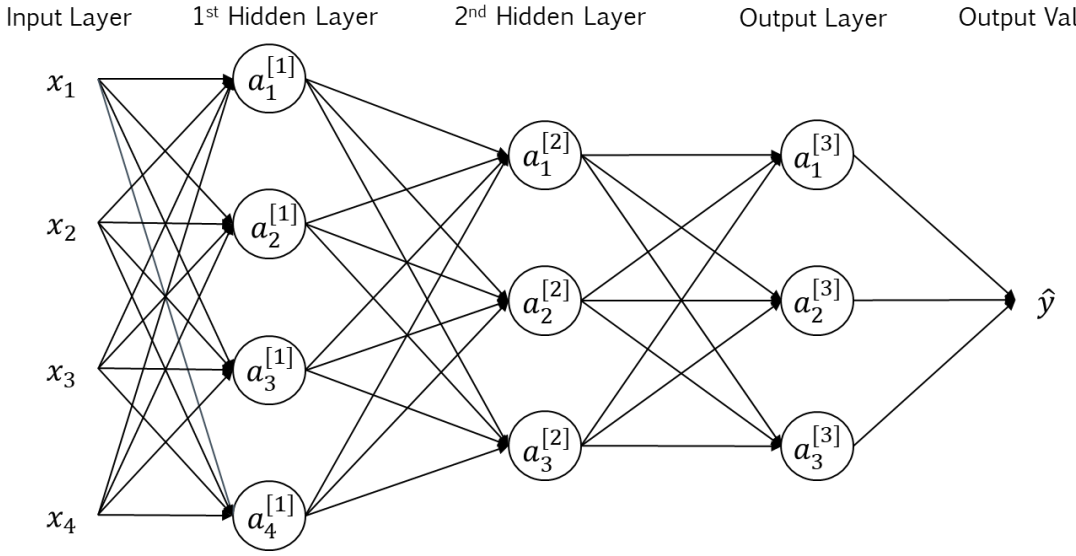


ANN Zoomed In

Binary Classifier:



3-class Classifier:

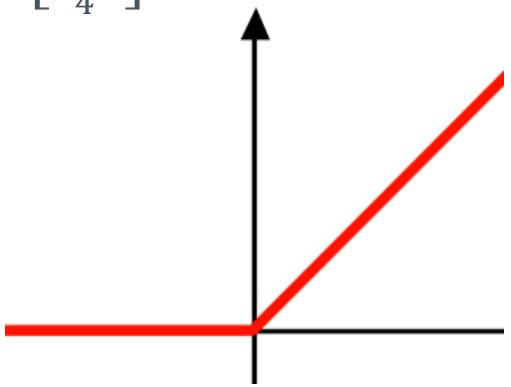


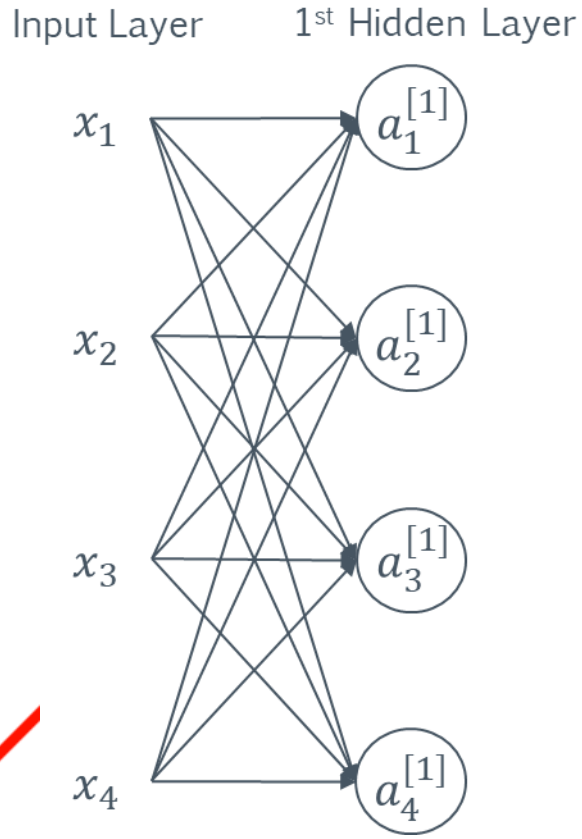
ANN Zoomed In: First Layer

$$z^{[1]} = W^{[1]}x + b$$

$$\begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \\ z_3^{[1]} \\ z_4^{[1]} \end{bmatrix} = \begin{bmatrix} w_{1,1}^{[1]} & w_{1,2}^{[1]} & w_{1,3}^{[1]} & w_{1,4}^{[1]} \\ w_{2,1}^{[1]} & w_{2,2}^{[1]} & w_{2,3}^{[1]} & w_{2,4}^{[1]} \\ w_{3,1}^{[1]} & w_{3,2}^{[1]} & w_{3,3}^{[1]} & w_{3,4}^{[1]} \\ w_{4,1}^{[1]} & w_{4,2}^{[1]} & w_{4,3}^{[1]} & w_{4,4}^{[1]} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ b_3^{[1]} \\ b_4^{[1]} \end{bmatrix}$$

$$a^{[1]} = RELU\left(z^{[1]}\right)$$

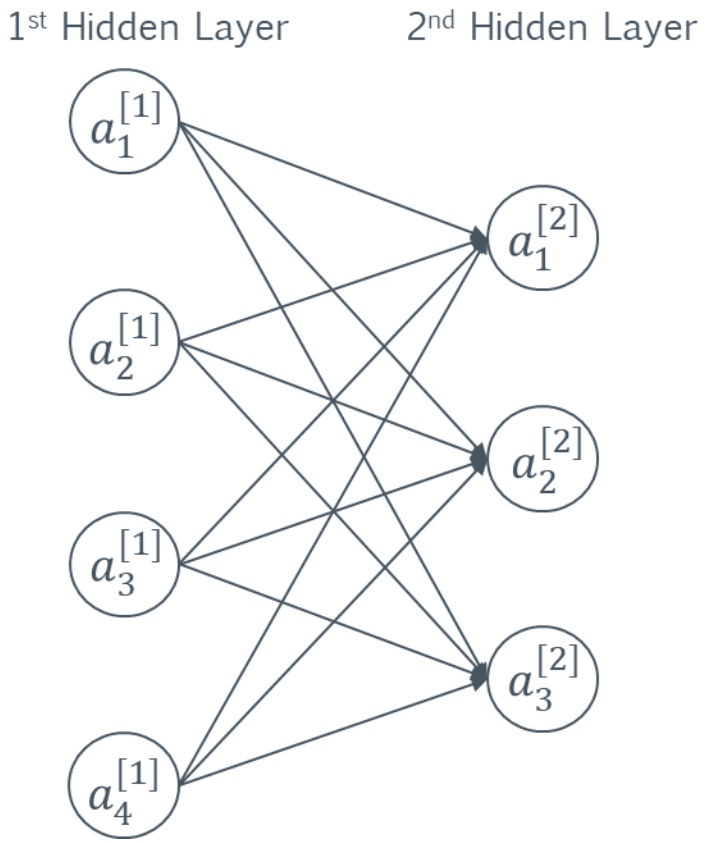




ANN Zoomed In: Second Layer

$$z^{[2]} = W^{[2]}a^{[1]} + b^{[2]}$$

$$\begin{bmatrix} z_1^{[2]} \\ z_2^{[2]} \\ z_3^{[2]} \end{bmatrix} = \begin{bmatrix} w_{1,1}^{[2]} & w_{1,2}^{[2]} & w_{1,3}^{[2]} & w_{1,4}^{[2]} \\ w_{2,1}^{[2]} & w_{2,2}^{[2]} & w_{2,3}^{[2]} & w_{2,4}^{[2]} \\ w_{3,1}^{[2]} & w_{3,2}^{[2]} & w_{3,3}^{[2]} & w_{3,4}^{[2]} \end{bmatrix} \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \\ a_3^{[1]} \\ a_4^{[1]} \end{bmatrix} + \begin{bmatrix} b_1^{[2]} \\ b_2^{[2]} \\ b_3^{[2]} \end{bmatrix}$$



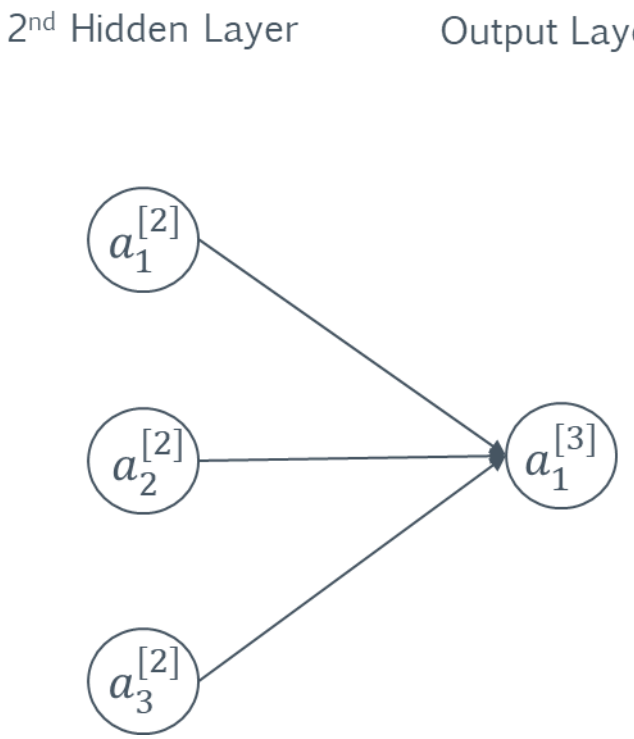
$$a^{[2]} = RELU\left(z^{[2]}\right)$$

ANN Zoomed In: Output Layer (Binary Classifier)

$$z^{[3]} = W^{[3]}a^{[2]} + b^{[3]}$$

$$\left[z_1^{[3]}\right] = \left[w_{1,1}^{[3]} \quad w_{1,2}^{[3]} \quad w_{1,3}^{[3]}\right]\begin{bmatrix} a_1^{[2]} \\ a_2^{[2]} \\ a_3^{[2]} \end{bmatrix} + \left[b_1^{[3]}\right]$$

$$a^{[3]} = Sig\left(z^{[3]}\right)$$



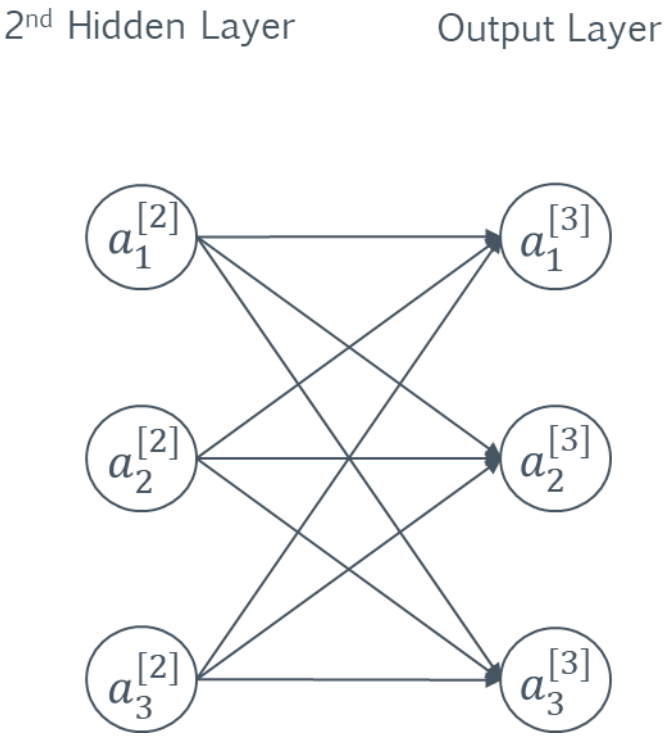
ANN Zoomed In: Output Layer (3-class Classifier)

$$z^{[3]} = W^{[3]}a^{[2]} + b^{[3]}$$

$$\begin{bmatrix} z_1^{[3]} \\ z_2^{[3]} \\ z_3^{[3]} \end{bmatrix} = \begin{bmatrix} w_{1,1}^{[3]} & w_{1,2}^{[3]} & w_{1,3}^{[3]} \\ w_{2,1}^{[3]} & w_{2,2}^{[3]} & w_{2,3}^{[3]} \\ w_{3,1}^{[3]} & w_{3,2}^{[3]} & w_{3,3}^{[3]} \end{bmatrix} \begin{bmatrix} a_1^{[2]} \\ a_2^{[2]} \\ a_3^{[2]} \end{bmatrix} + \begin{bmatrix} b_1^{[3]} \\ b_2^{[3]} \\ b_3^{[3]} \end{bmatrix}$$

$$a^{[3]} = softmax\left(z^{[3]}\right)$$

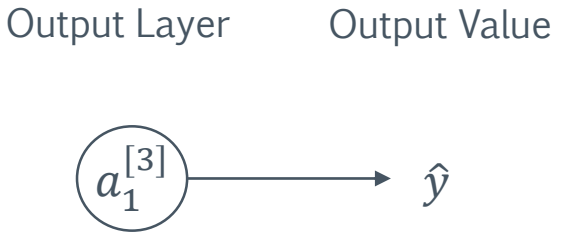
$$softmax(z) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$



ANN Zoomed In: Output Value

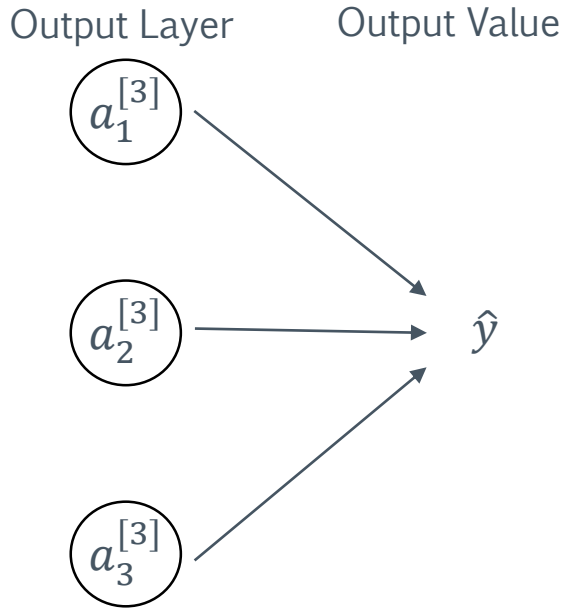
Binary Classifier:

$$\hat{y} = \begin{cases} 1, & a_1^{[3]} \geq T \\ 0, & a_1^{[3]} < T \end{cases}$$



Multiple classes:

$$\hat{y} = argmax\left(a^{[3]}\right)$$



Backpropagation

- Gradient Descent
  - Need to calculate partial derivative w.r.t each weight
- Backpropagation:
  - Propagates derivatives from output layer backwards
    - Chain Rule
- Unlike logistic regression, cost function is not convex or smooth
  - Gradient descent isn't guaranteed to find global minimum

Hyperparameters

- Gradient descent only learns weight matrices and bias vectors
  - Every other parameter is called a hyperparameter and must be chosen by the user

- Hyperparameters include:
  - How many layers
  - Number of nodes in each layer
  - Learning rate
  - Etc.

Applying a Neural Network to an image

- Need to convert image to column vector
  - Loss of local connectivity

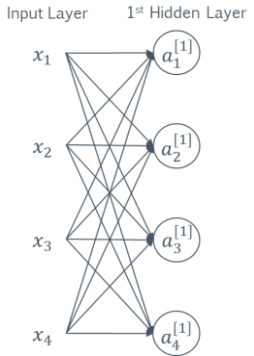
- In a fully connected network, each pixel is connected to each node in the hidden layer
  - Results in huge weight matrix
  - EX: With only 100 hidden neurons a CT input image would have over 26M weights in the first layer!

- Fully connected networks are very expensive to train for image classification

Transition from NN to CNN

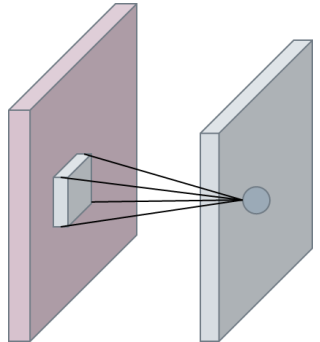
NEURAL NETWORK

- Uses matrix multiplication to calculate input to activation function
- $z = Wx + b$
- x is column vector
- Global connectivity



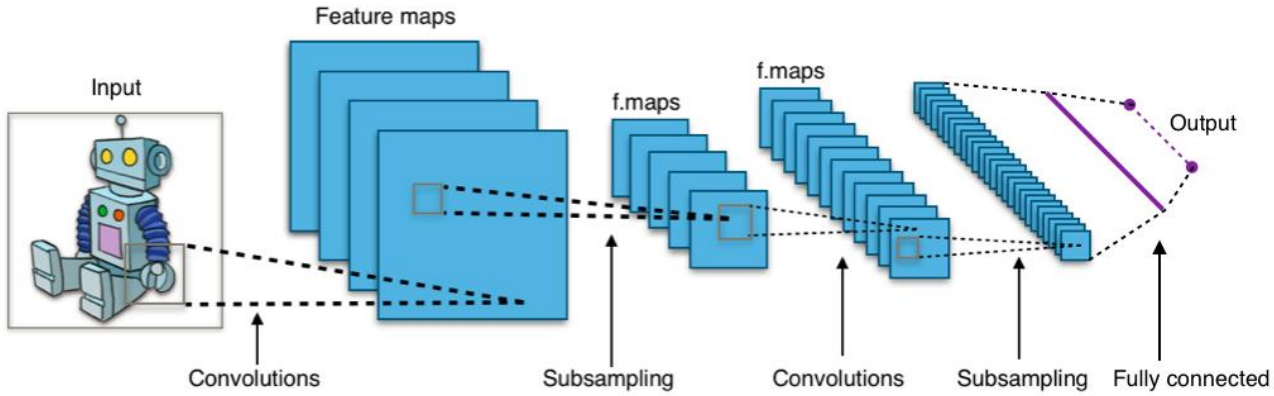
CNN

- Uses convolution to calculate input to activation function
- $Z = W * X + b$
- X is image matrix
- Local connectivity
- Parameter sharing

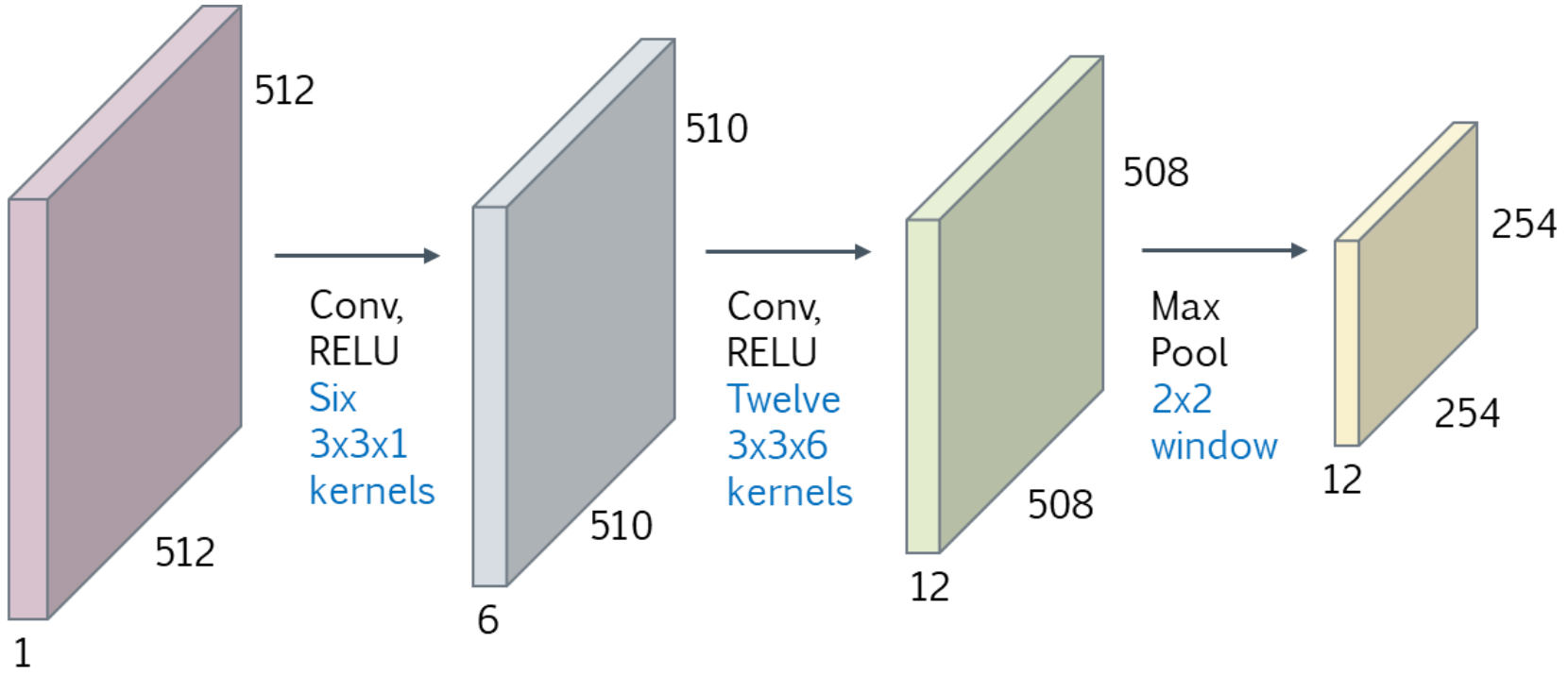


Convolutional Neural Network

- Typically outperforms neural network architecture for image based tasks
- Composed of a combination of convolutional layers and pooling layers followed by several fully connected layers
  - Max pooling introduces local invariance
- Convolution kernels are updated and learned based on backpropagation



CNN Zoomed In



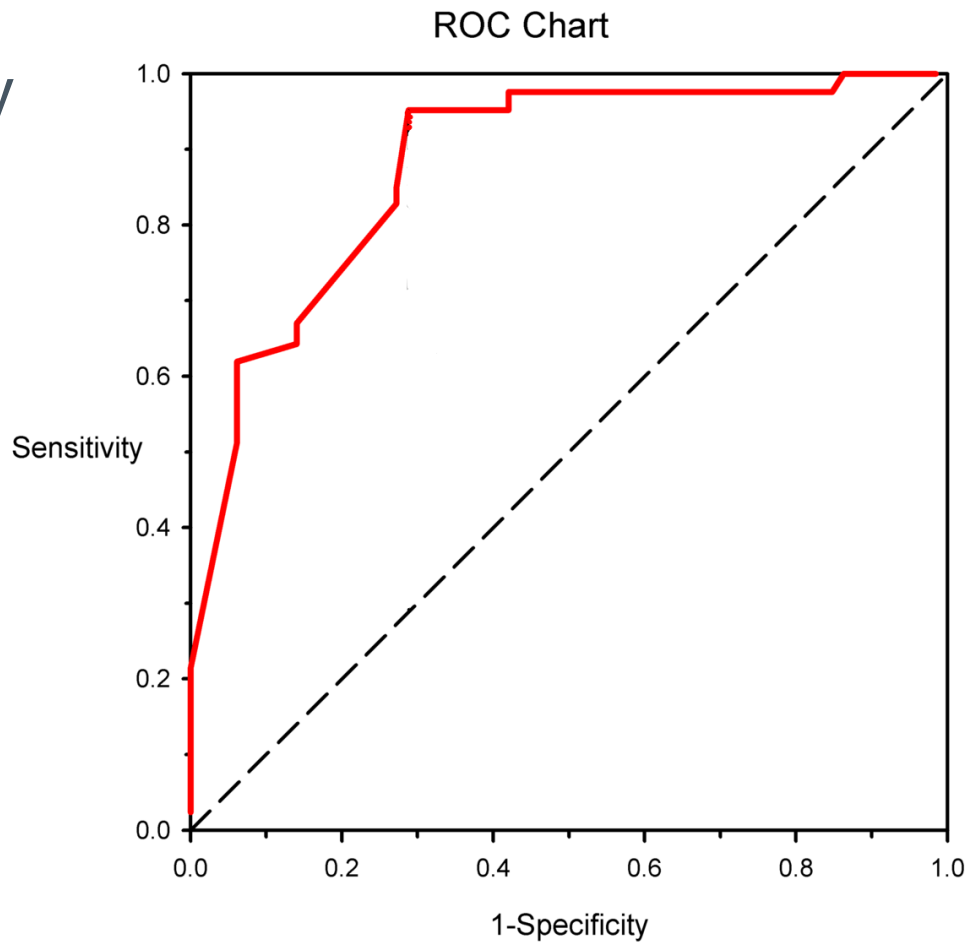
Performance Evaluation

- Need unbiased estimate of performance
  - 80/20 training and test set split
- Accuracy is not a tell all metric
  - Ex: CNN used to detect breast cancer in mammography images
    - ~95% of images are negative
    - CNN that just outputs negative will have 95% accuracy
- Sensitivity or specificity alone don't suffice either
  - Need metric that can account for possible class imbalance
- F_1 score combines Sensitivity and Precision

$$F_1 = 2 \cdot \frac{precision \times sensitivity}{precision + sensitivity}$$

Receiver Operating Characteristic (ROC) Curve

- Plots Sensitivity vs 1-Specificity
- Produced by varying threshold value T
- AUC = Area Under the Curve



Applications in Medical Imaging

Diagnostic:

› Computer Aided Diagnosis (CAD)

› Image reconstruction

› Automated segmentation

› Workflow Optimization

Therapy:

› Automated contouring

› Dose map calculations

End