Neural Network Basics

CS4742 Natural Language Processing
Lecture 05

Jiho Noh

Department of Computer Science
Kennesaw State University

CS4742 Summer 2025 [1]

KENNESAW STATE UNIVERSITY
COLLEGE OF COMPUTING AND SOFTWARE ENGINEERING

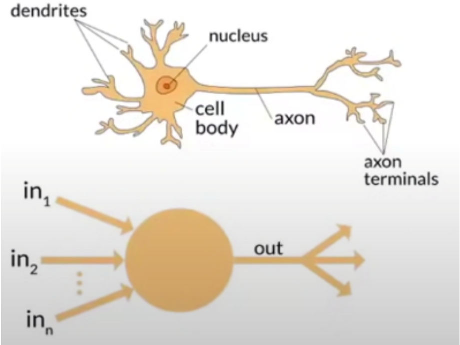
[1] This lecture is based on the slides from Dr. Hafiz Khan at KSU.

Topics

Where is deep learning the best-known approach?

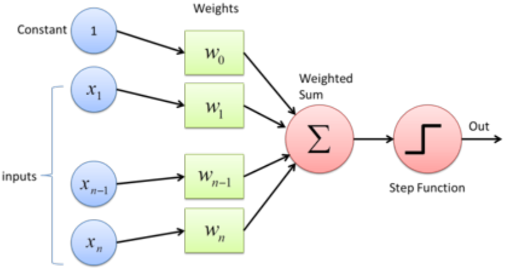
NLP	answering questions; speech recognition; summarizing documents; classifying documents; finding names, dates, etc. in documents; searching for articles mentioning a concept
Computer vision	satellite and drone imagery interpretation (e.g. for disaster resilience); face recognition; image captioning; reading traffic signs; locating pedestrians and vehicles in autonomous vehicles
Medicine	finding anomalies in radiology images, including CT, MRI, and x-ray; counting features in pathology slides; measuring features in ultrasounds; diagnosing diabetic retinopathy
Biology	folding proteins; classifying proteins; many genomics tasks, such as tumor-normal sequencing and classifying clinically actionable genetic mutations; cell classification; analyzing protein/protein interactions
Image generation	colorizing images; increasing image resolution; removing noise from images; converting images to art in the style of famous artists
Recommendation systems	web search; product recommendations; home page layout
Playing games	better than humans and better than any other computer algorithm at Chess, Go, most Atari videogames, many real-time strategy games
Robotics	handling objects that are challenging to locate (e.g. transparent, shiny, lack of texture) or hard to pick up
Other applications	financial and logistical forecasting; text to speech; much much more...

Neural Networks



1943 Warren McCulloch proposed a mathematical model of an artificial neuron

Deep learning is just a multiple layers of Neural Network Learning, "Deep Learning"



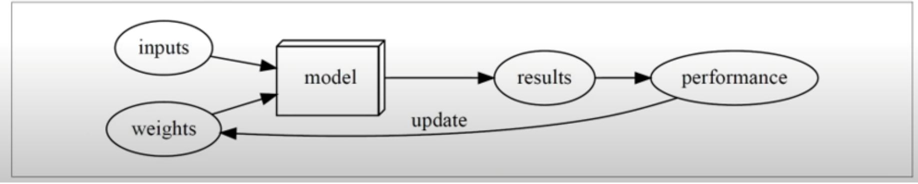
Human Brain

- Brain
  - Composed of a very large (10^{11}) number of processing units, *neurons*, operating in **parallel**.
  - Neurons in the brain have connections, called *synapses*.
  - The large connectivity is the power.
  - It is believed that both the processing and memory are distributed together over the network.
- Artificial Neural Network (ANN) models
  - Our aim is to utilize the function of brain to build useful machine.

NN as a paradigm for Parallel Processing

- Neural Networks (NN) are a class of models that are built with layers. Commonly used types of neural networks include convolutional and recurrent neural networks.
- Learning
  - Distribute a task over a network of small processors and to determine the local parameter values.
- No need to program and determine the parameter values ourselves; Such machines can learn from examples.
- Graphical Processing Unit (GPU) is good at doing this kind of job.

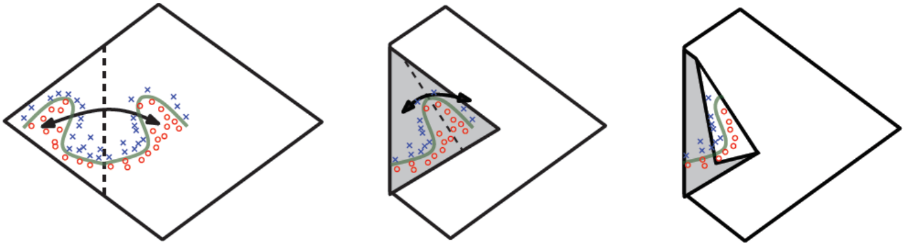
Schema of training a deep learning model



1. inputs: Preprocess data to be used for training, validation, testing.
2. models and weights: define the architecture of the neural network, and initialize the learnable weights.
3. performance assessment: compare predictions to actual values using a loss function to measure error.
4. optimization: update the weights using an optimization algorithm (like gradient descent) to minimize the loss.
5. iterate the above process

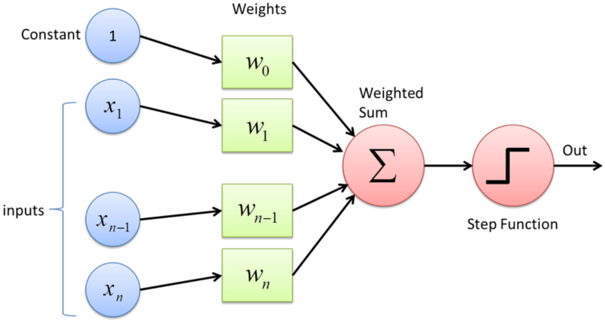
"Deep" Learning

- **Universal Approximation Theorem** stats that a neural network with one hidden layer can approximate any continous function on a compact domain, given sufficient neurons.
- *So why deeper?*
  - Shallow net may need (exponentially) more width.
  - Shallow net may overfit more



Topics

Perceptron Model

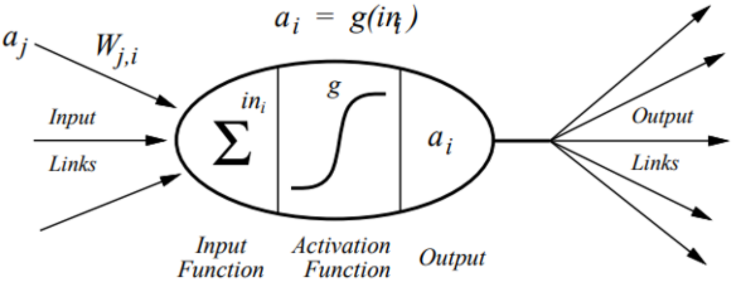


$$y = \sum_{j=1}^{d} w_j x_j + w_0$$

- Or, write the output as a dot product.

$$y = w^T x$$

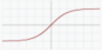
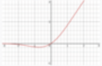
Non-linearity



$$a_i = g(\sum_j W_{j,i} a_j)$$

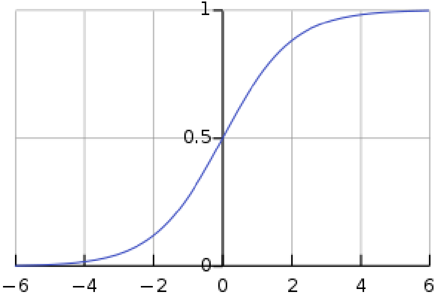
Activation Function

- An activation function decides whether a neuron should be activated and transmit a signal to the next connected neuron.
- That is, an activation function computes how important the set of input signals is.
- Activation functions are used at the end of a hidden unit to introduce non-linear comlexities to the model.
- NN without the activation functions?
  - Every neuron will be performing a linear transformation.
  - The composition of two linear functions is a linear function itself; the model would be just a linear regression model.
  - NN will loose higher abstraction capability.

Name	Plot	Function, $f(x)$	Derivative of f, $f'(x)$	Range
Identity		x	1	$(-\infty, \infty)$
Binary step		$\begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$	$\begin{cases} 0 & \text{if } x \neq 0 \\ \text{undefined} & \text{if } x = 0 \end{cases}$	$\{0, 1\}$
Logistic, sigmoid, or soft step		$\sigma(x) = \dfrac{1}{1 + e^{-x}}$	$f(x)(1 - f(x))$	$(0, 1)$
Hyperbolic tangent (tanh)		$\tanh(x) = \dfrac{e^x - e^{-x}}{e^x + e^{-x}}$	$1 - f(x)^2$	$(-1, 1)$
Rectified linear unit (ReLU)[7]		$\begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$ $= \max\{0, x\} = x\mathbf{1}_{x>0}$	$\begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x > 0 \\ \text{undefined} & \text{if } x = 0 \end{cases}$	$[0, \infty)$
Gaussian Error Linear Unit (GELU)[4]		$\dfrac{1}{2}x\left(1 + \text{erf}\left(\dfrac{x}{\sqrt{2}}\right)\right)$ $= x\Phi(x)$	$\Phi(x) + x\phi(x)$	$(-0.17\ldots, \infty)$
Softplus[8]		$\ln(1 + e^x)$	$\dfrac{1}{1 + e^{-x}}$	$(0, \infty)$
Exponential linear unit (ELU)[9]		$\begin{cases} \alpha(e^x - 1) & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$ with parameter α	$\begin{cases} \alpha e^x & \text{if } x < 0 \\ 1 & \text{if } x > 0 \\ 1 & \text{if } x = 0 \text{ and } \alpha = 1 \end{cases}$	$(-\alpha, \infty)$
Scaled exponential linear unit (SELU)[10]		$\lambda\begin{cases} \alpha(e^x - 1) & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$ with parameters $\lambda = 1.0507$ and $\alpha = 1.67326$	$\lambda\begin{cases} \alpha e^x & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$	$(-\lambda\alpha, \infty)$
Leaky rectified linear unit (Leaky ReLU)[11]		$\begin{cases} 0.01x & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$	$\begin{cases} 0.01 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$	$(-\infty, \infty)$
		$\begin{cases} \alpha x & \text{if } x < 0 \\ \ldots \end{cases}$	$\begin{cases} \alpha & \text{if } x < 0 \\ \ldots \end{cases}$	

Common Activation Functions — Sigmoid

$$g(x) = \frac{1}{1 + e^{-x}}$$

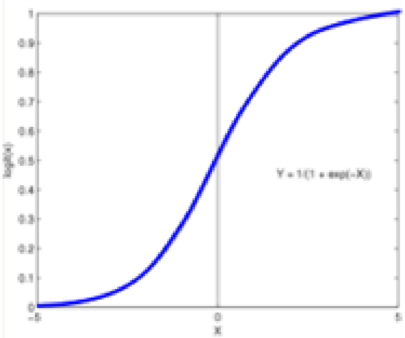


Logistic Regression

Assumes the following functional form for $P(Y|X)$:

$$P(Y = 1|X) = \frac{1}{1 + exp(-(w_0 + \sum_i w_i X_i))}$$

Logistic function applied to a linear function of the data

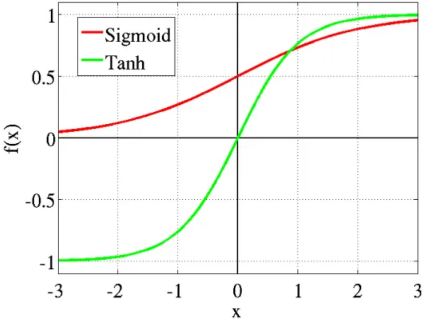


Logistic function (or Sigmoid)

$$\frac{1}{1 + exp(-z)}$$

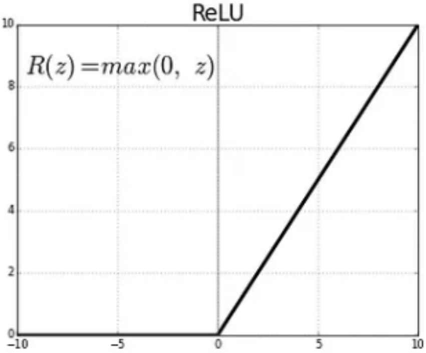
Common Activation Functions — Tanh

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



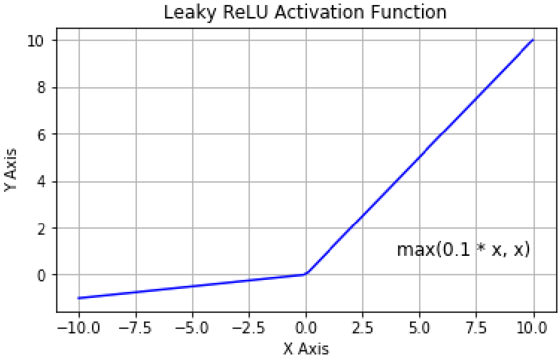
Common Activation Functions — ReLU

$$R(x) = \max(0, z)$$



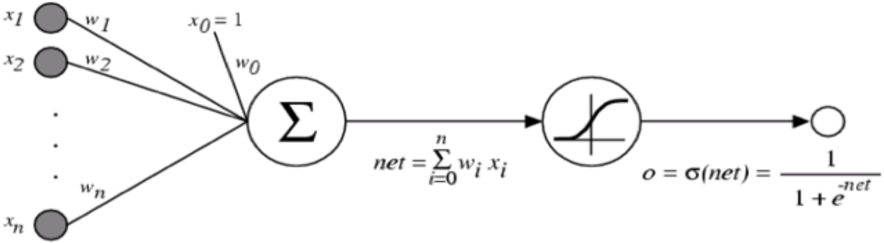
Common Activation Functions — Leaky ReLU

$$g(x) = \max(\epsilon z, z) \text{ with } \epsilon << 1$$



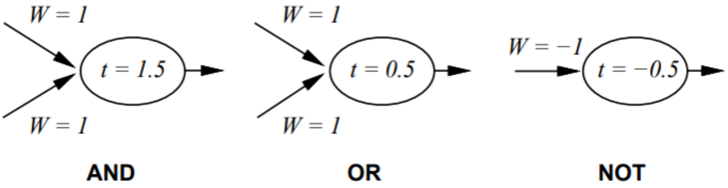
Logistic function as a Graph

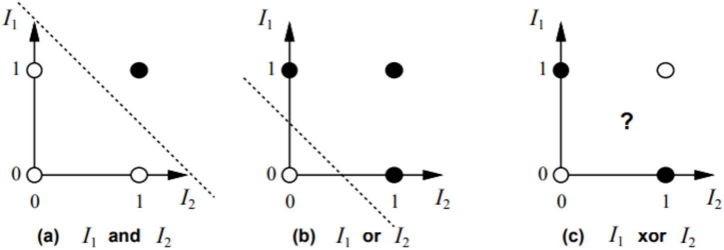
Output, $o(x) = \sigma(w_0 + \sum_i w_i X_i) = \frac{1}{1+exp(-(w_0+\sum_i w_i X_i))}$



Neural networks represent a function f by network of logistic/sigmoid units

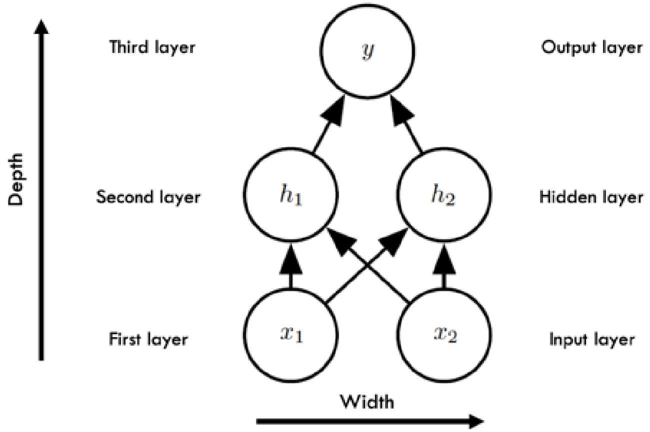
Boolean Functions and Perceptron



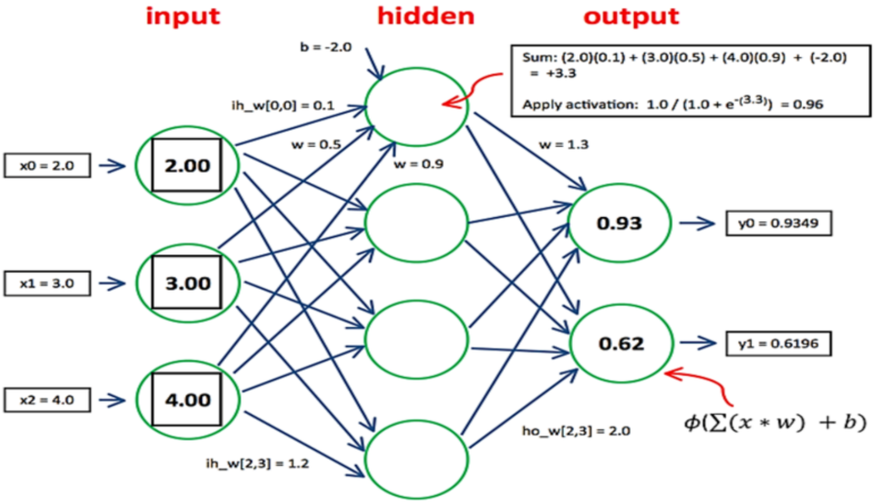


Neural Networks

- Feedforward Networks or Multilayer Perceptrons (MLPs)
- $y = f^*(x; \vartheta)$ maps an input x to y



Feedforward Network Example

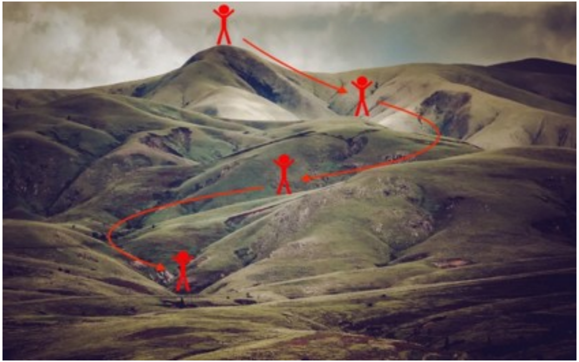


Topics

NN Learning as Optimization

- NN learning is cast as an optimization (search) problem.
- Navigate the space of model weights in order to make good predictions.

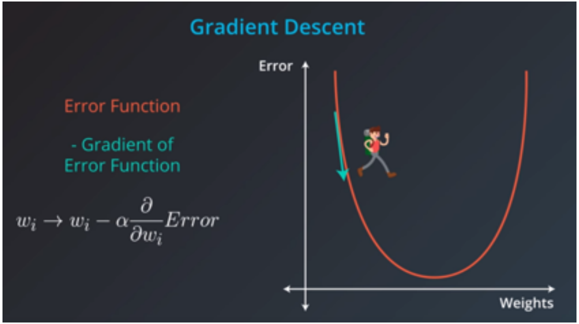
- How do we update the weights to improve at any given task?



Stochastic Gradient Descent

- Update weights using the backpropagation of error.

- (Error) Gradient
  - *Gradient Descent* algorithm seeks to update the model weights to reduce the next evaluation error
  - Optimization algorithm navigates down the gradient (or slope) of error.



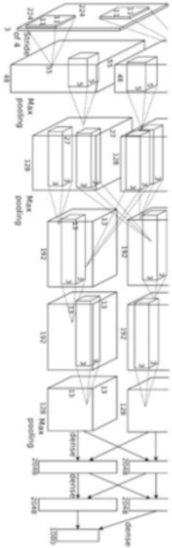
- Stochastic means "random"
  - Randomness in selecting a training example.
  - Incrementally update weights using each training example.

Convolutional network (AlexNet)

input image

weights

loss



Loss Functions

- *The function that computes the distance between the current output of the algorithm and the expected output.*

- Regression
  - Mean Squared Error Loss
  - Mean Squared Logarithmic Error Loss
  - Mean Absolute Error Loss
- Binary Classification
  - Binary Cross-Entropy
  - Hinge Loss
  - Squared Hinge Loss
- Multi-class Classification
  - Multi-class Cross-Entropy Loss
  - Sparse Multiclass Cross-Entropy Loss
  - Kullback-Leibler Divergence loss

Loss Functions

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

$$MSLE = \frac{1}{n}\sum_{i=1}^{n}\left(\log Y_i - \log \hat{Y}_i\right)^2$$

$$BCE = \frac{1}{n}\sum_{i=1}^{n} Y_i \log \hat{Y}_i + (1 - Y_i)\log(1 - \hat{Y}_i)$$

$$Hinge = \frac{1}{n}\sum_{i=1}^{n} max(0, 1 - Y_i\hat{Y}_i)$$

$$KL = \frac{1}{n}\sum_{i=1}^{n} Y_i \log(Y_i/\hat{Y}_i)$$

Topics

1. **Deep Learning**
2. **Multilayer Perceptron**
3. **Learning**
   - Backprogagation

Backpropagation

- Dr. Fei-Fei Li's Backpropagation Lecture Slides

Summary