CS/ENGR M148 L14:
Introduction to Neural Networks

Sandra Batista

Administrative News

This week in discussion section:

In lab this week you'll be learning about PyTorch to apply neural networks to your projects.

No project check-ins this week.

<u>Lecture on 11/25/24 will be via Zoom for the holiday.</u>

Midterm grades posted. Grade distribution on piazza. Regrade requests due by 6 pm 11/20/24.

<u>PS3 due today and quiz Thursday.</u>

<u>Will post final project report guidelines this week also with PS4.</u>

Administrative News

Extra credit Final Exam Review Question Code Bank:

https://forms.gle/XdC97wxwWd8QuTR9A

Questions due by 11:59 pm PT on 11/25/24.

We'll share questions with solutions during week 10 for final exam review.

Join our slido for the week…

https://app.sli.do/event/nCV57u4mC7eUMit9euSBr2

Today's Learning Objectives

Students will be able to:

- Explain how Expectation Maximization (EM) is used for Clustering: Gaussian Mixture Models
- Apply the Forward and Backwards Algorithms to Hidden Markov Models (HMM)
- Describe what a neural network is
- Apply the forward algorithm on the MNIST NN

Expectation Maximization (EM)

- Enables parameter estimation (learning) in probabilistic models with incomplete data.
- Uses Maximum Likelihood Estimation (MLE).
- MLE is a way to assess the quality of a statistical model based on the probability that model assigns to the observed data. The model that has the highest probability of generating the data is the best one.
- EM algorithm is used to find (locally) MLE parameters of a statistical model in cases where the equations cannot be solved directly.

[Shah 2020]

EM Experiment

- We are given 2 biased coins, but we don't know what probability of heads are for each. How can we learn this?

[Shah 2020]

EM Experiment

- What are **maximum likelihood estimates?**

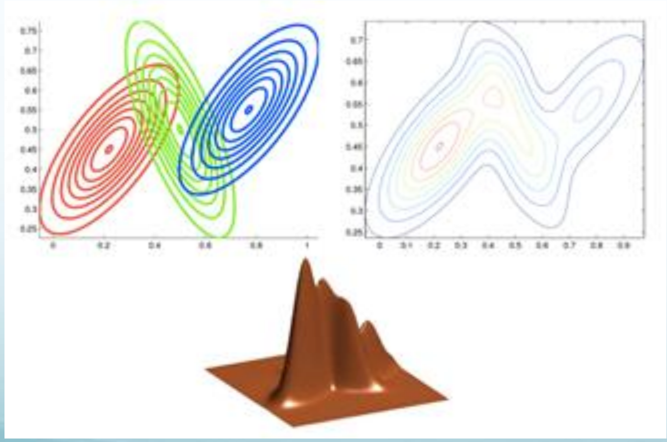
- What if we don't know what coin was tossed?

Gaussian Mixture Models

- A Gaussian mixture model is a family of distributions of the form

$$p(\mathbf{x}) = \sum_{i=1}^{k} \pi_i \mathcal{N}(\mathbf{x}|\mu_i, \Sigma_i).$$

- The π_i are *mixing coefficients* that sum to 1.
- Estimate the mean and covariance matrices from data as parameters for GMM from data
- Each Gaussian distribution represents a cluster.
- This provides a generative model for clustering.



Zemel et al., 2016

Gaussian Mixture Model

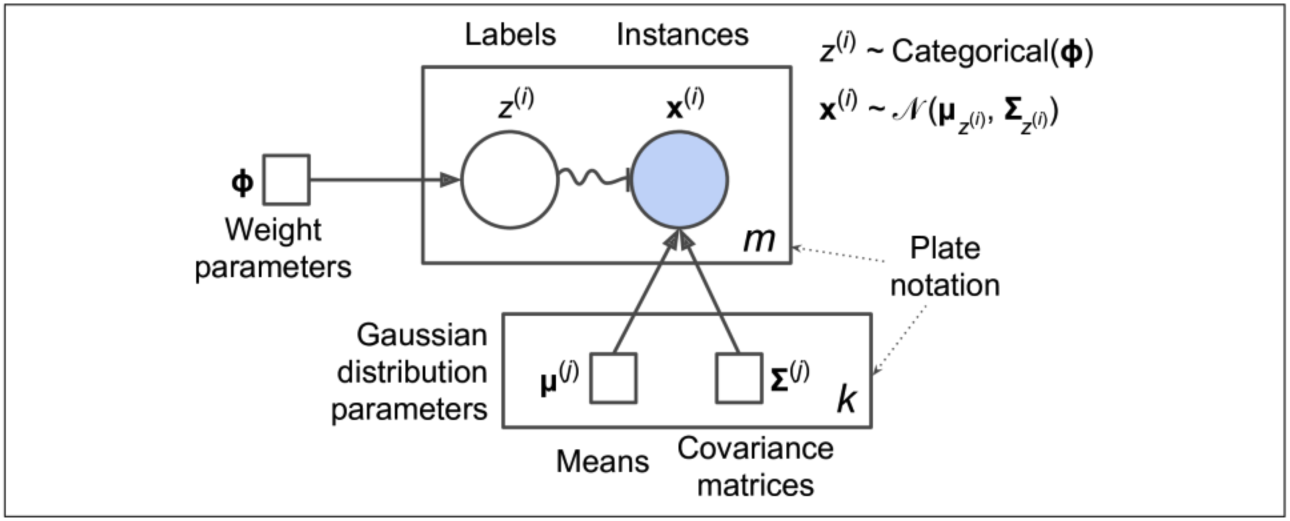


Figure 9-16. Gaussian mixture model

Aurelien Geron. 2019. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2nd. ed.). O'Reilly Media, Inc

Maximum likelihood estimation

- The log likelihood function that must be maximized is:

$$\log Pr(x|\pi,\mu,\sigma) = \sum_{j=1}^{n} \log\left(\sum_{i=1}^{k} \pi N(x|\mu_i,\Sigma_i)\right)$$

- Expectation Step: Calculate clusters
- Maximization Step: Update parameters

Shortcoming of EM algorithm for clustering

- How to choose the number of clusters (k)?
- The EM algorithm can get stuck in local maxima; initialization is corncer
- EM algorithm may be slow.
- If data not from a Gaussian mixture process, EM algorithm may converge to a poor solution.

Your turn:
GMM on GPU data

Please get the Jupyter notebook for GPU data:

Go to:

The data file on BruinLearn Week 7 Module:

- `sgemm_product.csv`

Notebook:
https://colab.research.google.com/drive/1iqVnrE7LKQyW_UXExzV3Q06EP10Q2Bvk?usp=sharing

[Shah 2020]

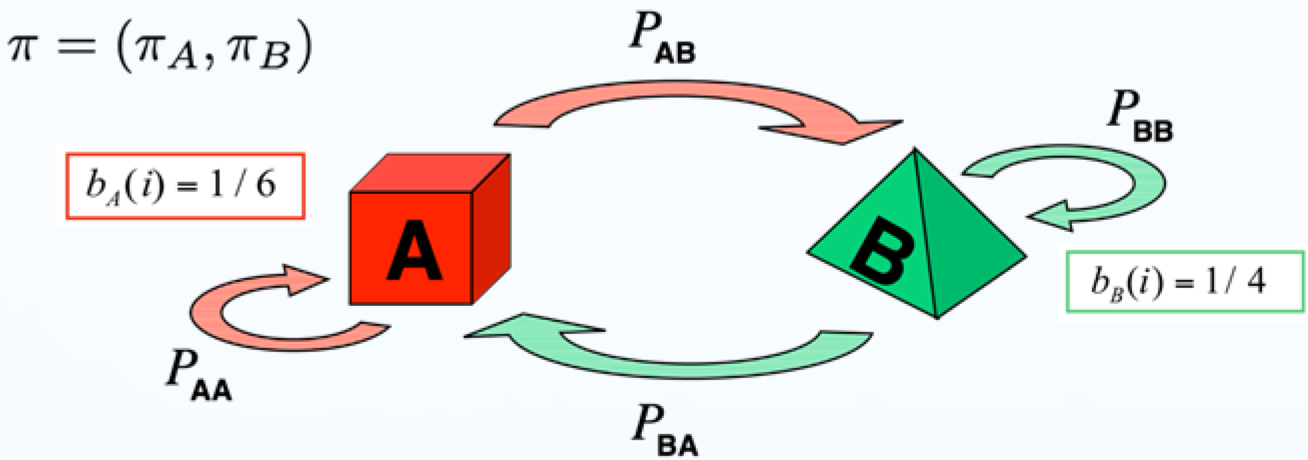
Save a copy to your Google Drive and keep notes there...

Today's Learning Objectives

Students will be able to:

- ✓ Explain how Expectation Maximization (EM) is used for Clustering: Gaussian Mixture Models
- Apply the Forward and Backwards Algorithms to Hidden Markov Models (HMM)
- Describe what a neural network is
- Apply the forward algorithm on the MNIST NN

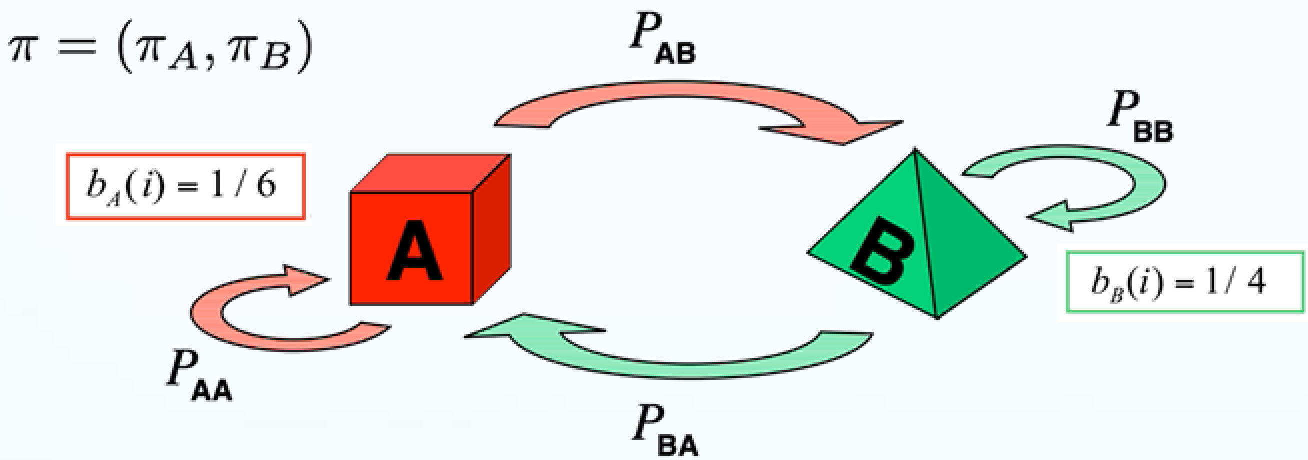
A hidden Markov model



$$\Pr(Y_0 = 1, Y_1 = 4, Y_2 = 3, Y_3 = 6, Y_4 = 6, Y_5 = 4) = ?$$

Solved with the forward algorithm.

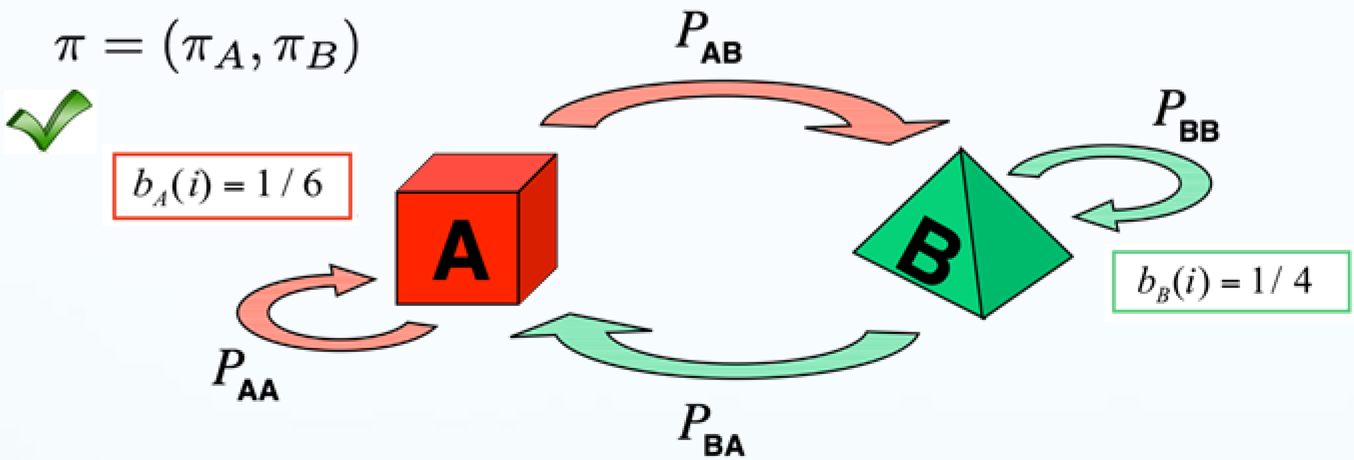
A hidden Markov model



What is the most likely sequence of states of the Markov chain to have resulted in 1,4,3,6,6,4?

Solved with the Viterbi algorithm.

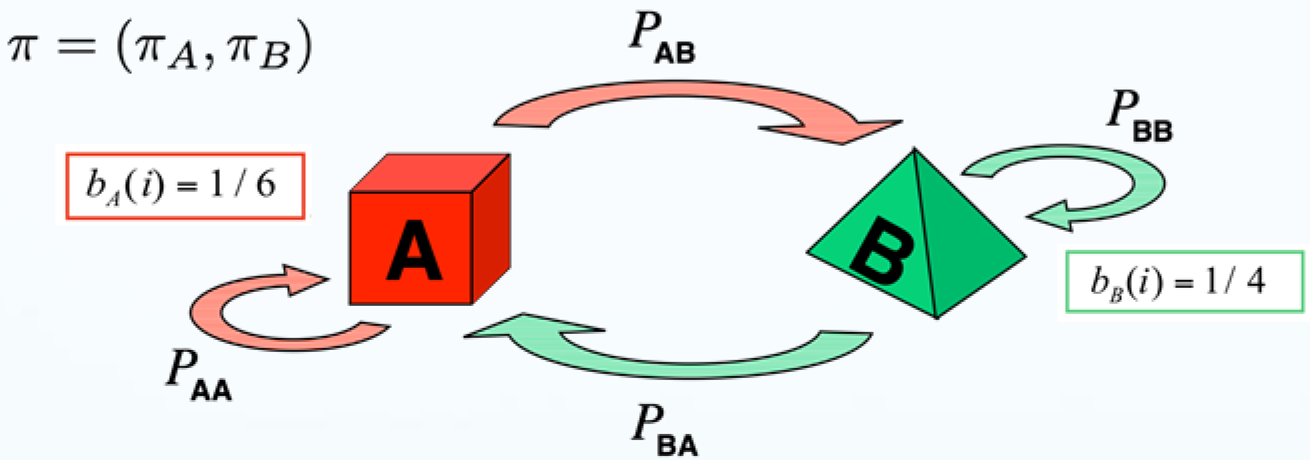
A hidden Markov model



What is the probability that $X_3 = B$ if $Y_0=1, Y_1=4, Y_2=3, Y_3=6, Y_4=6, Y_5=4$?

Solved with the forward-backward algorithm.

A hidden Markov model



*Given multiple sequences of numbers (observations of **Y**), estimate parameters for the model*

This is expectation-maximization algorithm

The occasionally dishonest casino

A casino uses a fair die most of the time, but occasionally switches to a loaded one

Fair die: Prob(1) = Prob(2) = . . . = Prob(6) = 1/6

Loaded die: Prob(1) = Prob(2) = . . . = Prob(5) = 1/10,

Prob(6) = ½

These are the ***emission*** probabilities

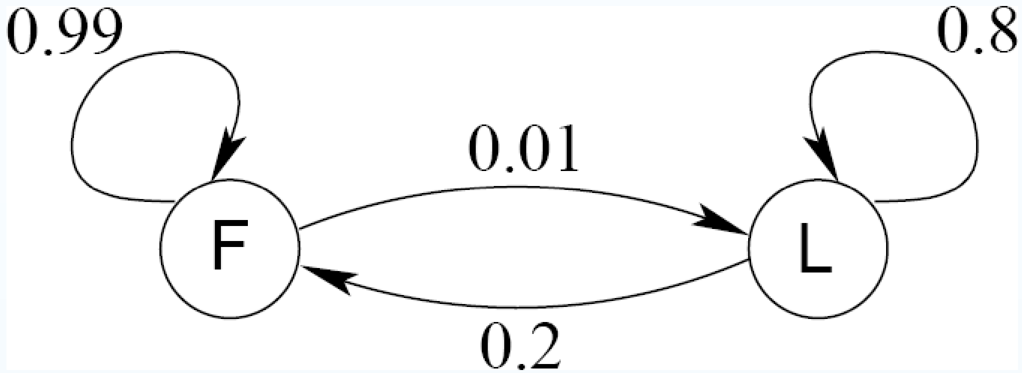
Transition probabilities

Prob(Fair –> Loaded) = 0.01

Prob(Loaded –>Fair) = 0.2

Transitions between states obey a Markov process

An HMM for the occasionally dishonest casino



Notation

- x *is the sequence of symbols emitted by model*
  - x_i *is the symbol emitted at time* i
- *A* ***path****,* π*, is a sequence of states*
  - *The* i*-th state in* π *is* π_i
- a_{kr} *is the probability of making a transition from state* k *to state* r*:*

$$a_{kr} = Pr(\pi_i = r \mid \pi_{i-1} = k)$$

- $e_k(b)$ *is the probability that symbol* b *is emitted when in state* k

$$e_k(b) = Pr(x_i = b \mid \pi_i = k)$$

The most probable path

The most likely path π^* satisfies

$$\pi^* = \underset{\pi}{\arg\max}\, Pr(x, \pi)$$

*To find π^*, consider all possible ways the last symbol of **x** could have been emitted*

Let

$$v_k(i) = \text{Prob. of path } \langle \pi_1, \cdots, \pi_i \rangle \text{ most likely}$$
$$\text{to emit } \langle x_1, \ldots, x_i \rangle \text{ such that } \pi_i = k$$

Then

$$v_k(i) = e_k(x_i) \max_r \left(v_r(i-1) a_{rk} \right)$$

The Viterbi Algorithm

- *Initialization* $(i = 0)$

$$v_0(0) = 1,\ \ v_k(0) = 0 \text{ for } k > 0$$

- *Recursion* $(i = 1, \ldots, L)$: *For each state* k

$$v_k(i) = e_k(x_i)\max_r\left(v_r(i-1)a_{rk}\right)$$

- *Termination:*

$$Pr(x, \pi^*) = \max_k\left(v_k(L)a_{k0}\right)$$

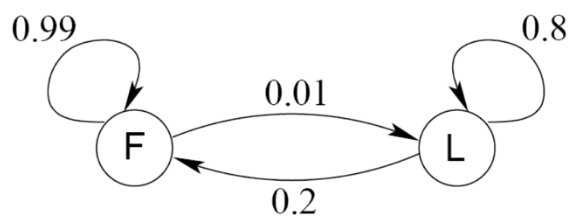
To find π^*, *use trace-back, as in dynamic programming*

Viterbi: Example

x

π	ε	6	2	6
B	**1**	**0**	**0**	**0**
F	**0**	(1/6)×(1/2) = **1/12**	(1/6)×max{(1/12)×0.99, (1/4)×0.2} = **0.01375**	(1/6)×max{0.01375×0.99, 0.02×0.2} = **0.00226875**
L	**0**	(1/2)×(1/2) = **1/4**	(1/10)×max{(1/12)×0.01, (1/4)×0.8} = **0.02**	(1/2)×max{0.01375×0.01, 0.02×0.8} = **0.08**

$$\boxed{v_k(i) = e_k(x_i)\max_r\left(v_r(i-1)a_{rk}\right)}$$



Total probabilty

Many different paths can result in observation x.

The probability that our model will emit x is

$$\Pr(x) = \sum_{\pi} \Pr(x, \pi)$$

Total Probability

If HMM models a family of objects, we want total probability to peak at members of the family. (Training)

Total probability

$\Pr(x)$ can be computed in the same way as probability of most likely path.

Let

$$f_k(i) = \text{Prob. of observing } \langle x_1, \ldots, x_i \rangle$$

$$\text{assuming that } \pi_i = k$$

Then

$$f_k(i) = e_k(x_i) \sum_r f_r(i-1) a_{rk}$$

and

$$\Pr(x) = \sum_k f_k(L) a_{k0}$$

The Forward Algorithm

- *Initialization* ($i = 0$)

$$f_0(0) = 1, \quad f_k(0) = 0 \text{ for } k > 0$$

- *Recursion* ($i = 1, \ldots, L$): *For each state* k

$$f_k(i) = e_k(x_i) \sum_r f_r(i-1) a_{rk}$$

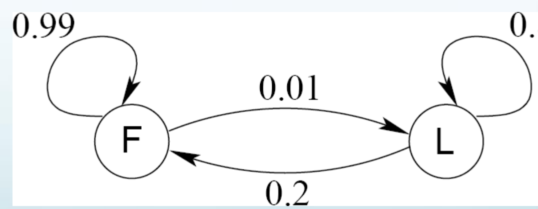
- *Termination:*

$$Pr(x) = \sum_k f_k(L) a_{k0}$$

Forward: Example

		x		

π	ε	6	2	6
B	1	0	0	0
F	0	$(1/6)\times(1/2) = 1/12$		
L	0	$(1/2)\times(1/2) = 1/4$		



The Backward Algorithm

- *Initialization (i = L)*

$$b_k(L) = a_{k0} \text{ for all } k$$

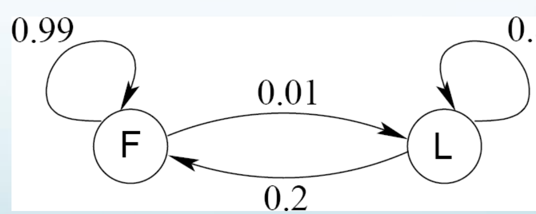
- *Recursion (i = L-1, . . . , 1): For each state k*

$$b_k(i) = \sum_l a_{kl} e_l(x_{i+1}) b_l(i+1)$$

Backward: Example

x

		ε	6	2	6
	B	1	0	0	0
π	F	0			
	L	0			



Backward Algorithm and Posterior Decoding

- *How likely is it that my observation comes from a certain state?*

$$P(x_i \text{ is emitted by state } k \mid \text{whole observation})$$

- *Like the Forward matrix, one can compute a Backward matrix*
- *Multiply Forward and Backward entries*

$$P(\pi_i = k \mid x) = \frac{f_k(i) \cdot b_k(i)}{P(x)}$$

  - *P(x) is the total probability computed by, e.g., forward algorithm*

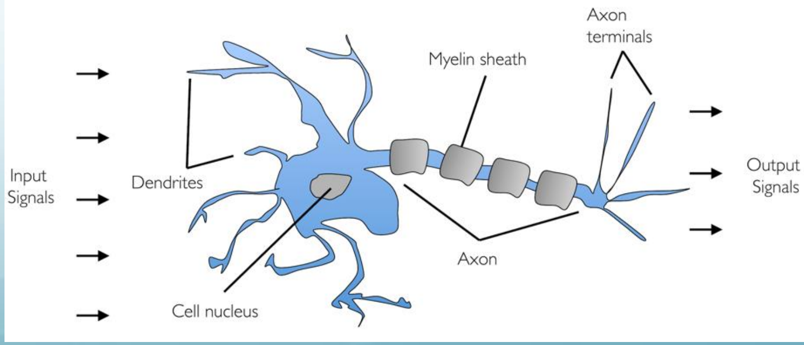
Today's Learning Objectives

Students will be able to:

- ✔ Explain how Expectation Maximization (EM) is used for Clustering: Gaussian Mixture Models
- ✔ Apply the Forward and Backwards Algorithms to Hidden Markov Models (HMM)
- Describe what a neural network is
- Apply the forward algorithm on the MNIST NN

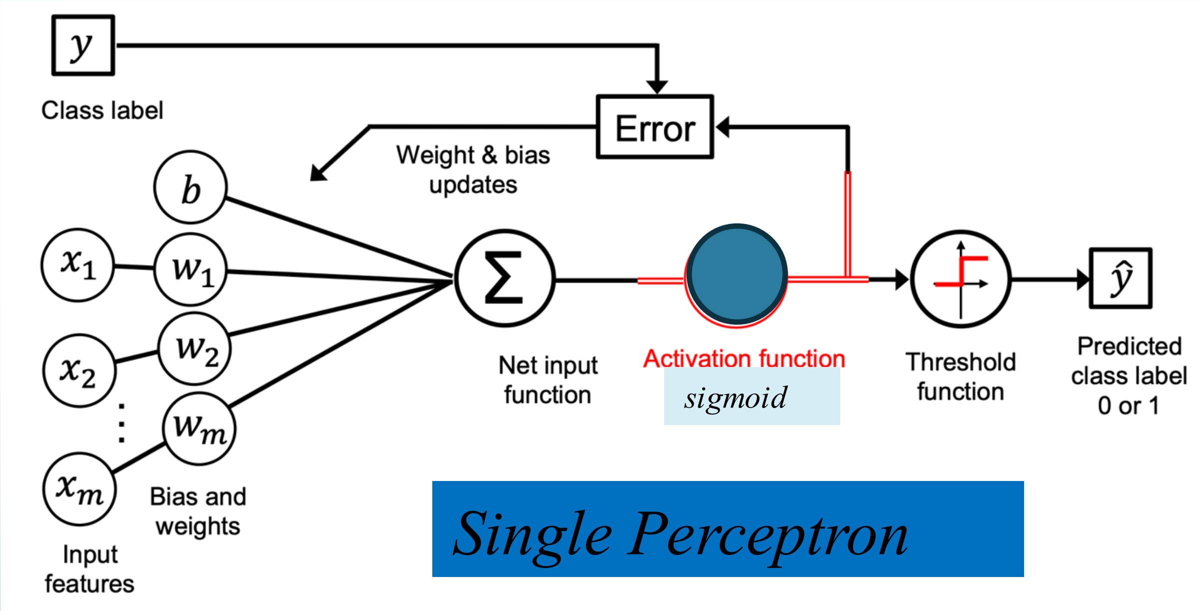
Neural Networks (NN)

- *Based on models for how brain works using artificial neurons*
- *Many varied successful applications such as mood recognition in pictures, modeling virus mutations, and predicting needed medical resources*
- *Used to model complex, nonlinear models*
- *We'll focus on using them for **classification**.*



What is a neural network?

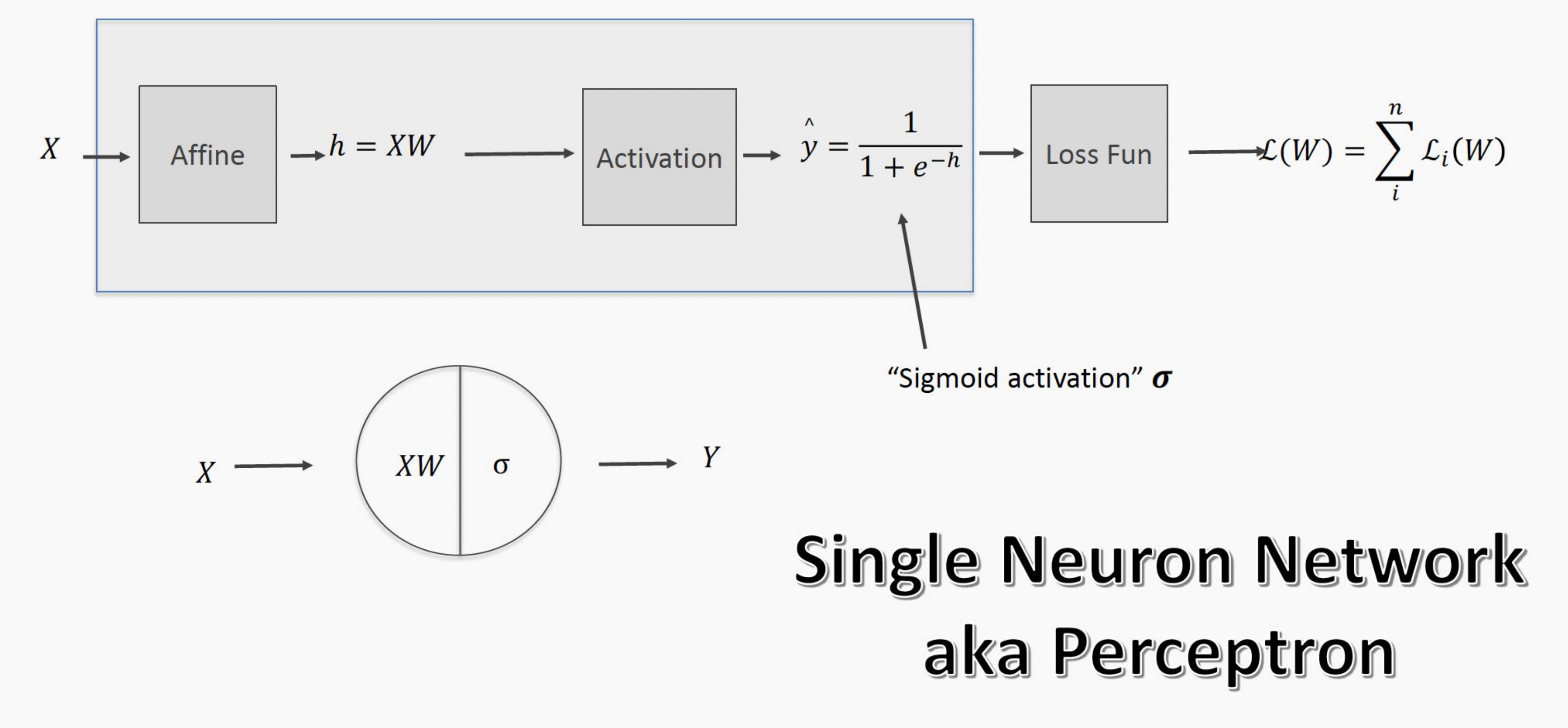
- *A **neural network** consists of layers of nodes or **artificial neurons***



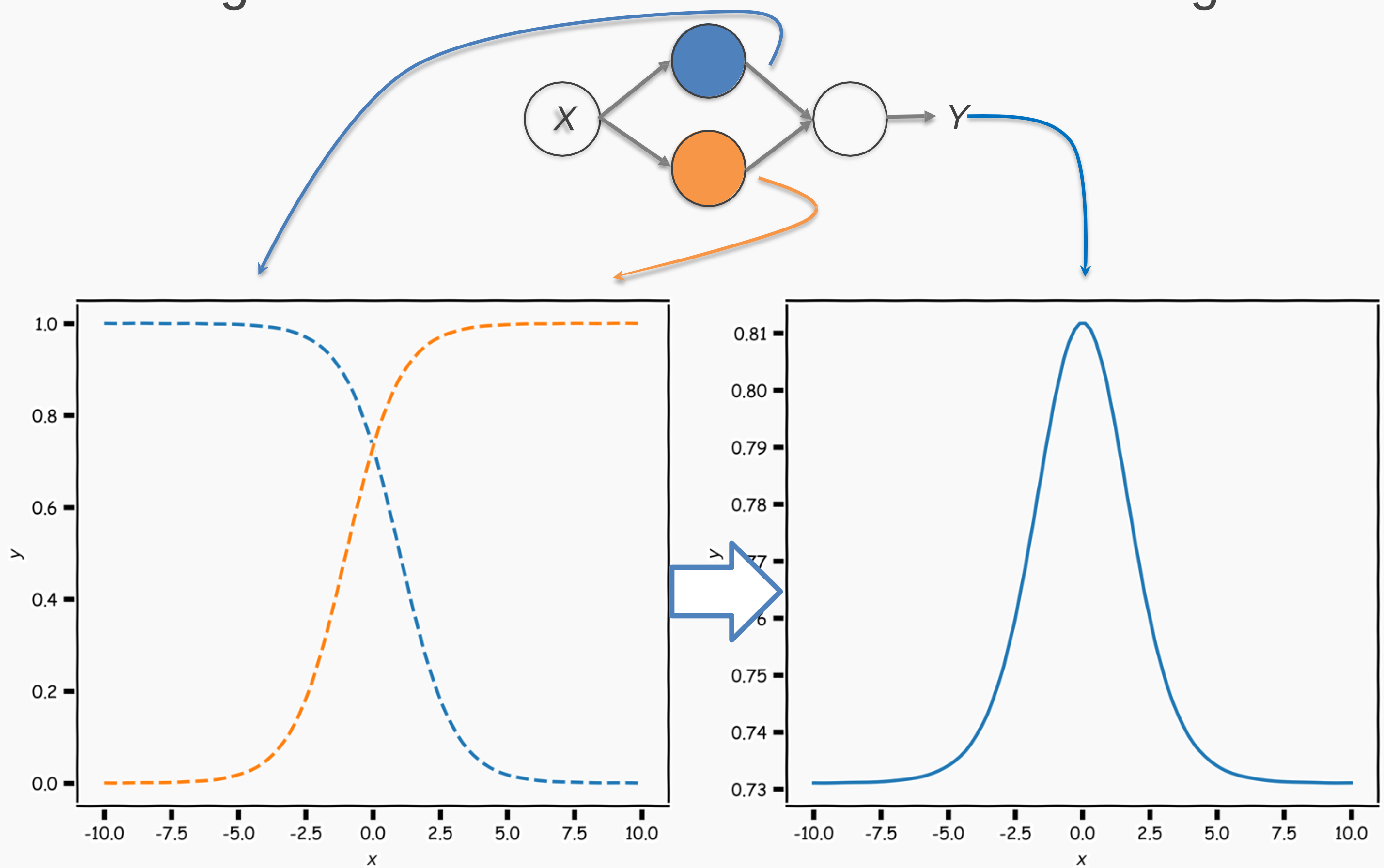
A single neuron can be a logistic regression or linear unit or other activation functions.

What is a neural network?

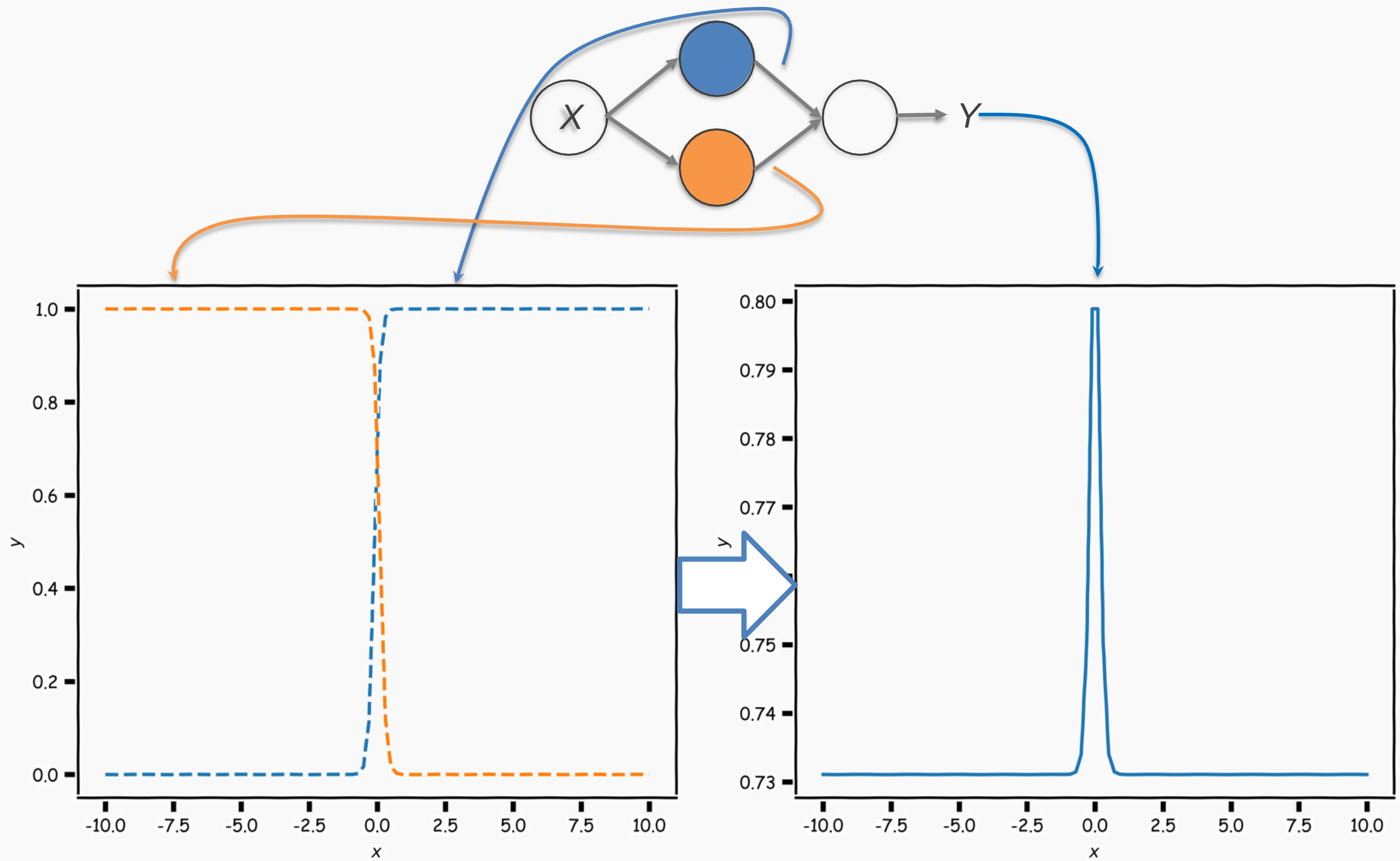
- *A **neural network** consists of layers of nodes or **artificial neurons***



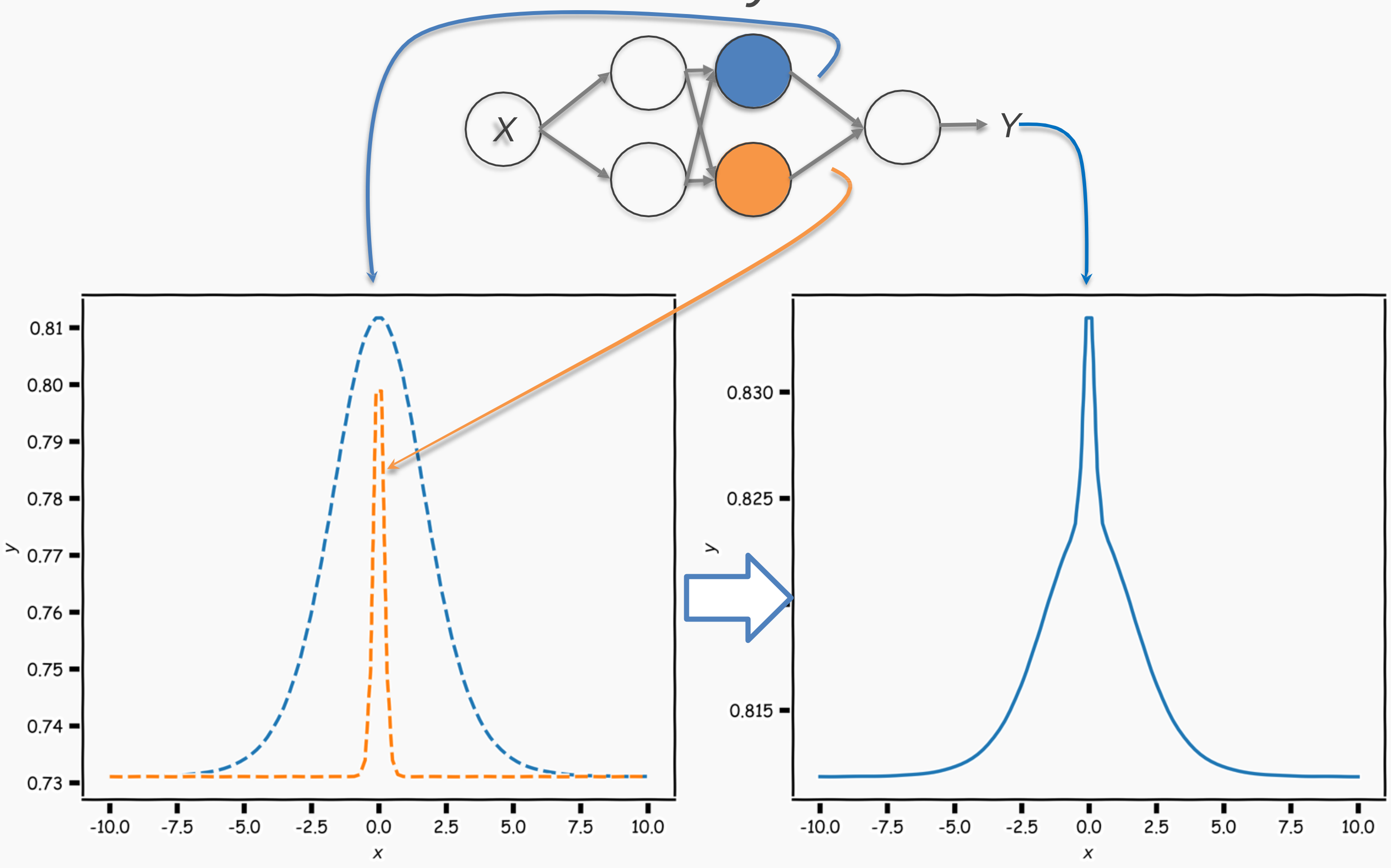
Combining neurons allows us to model interesting functions



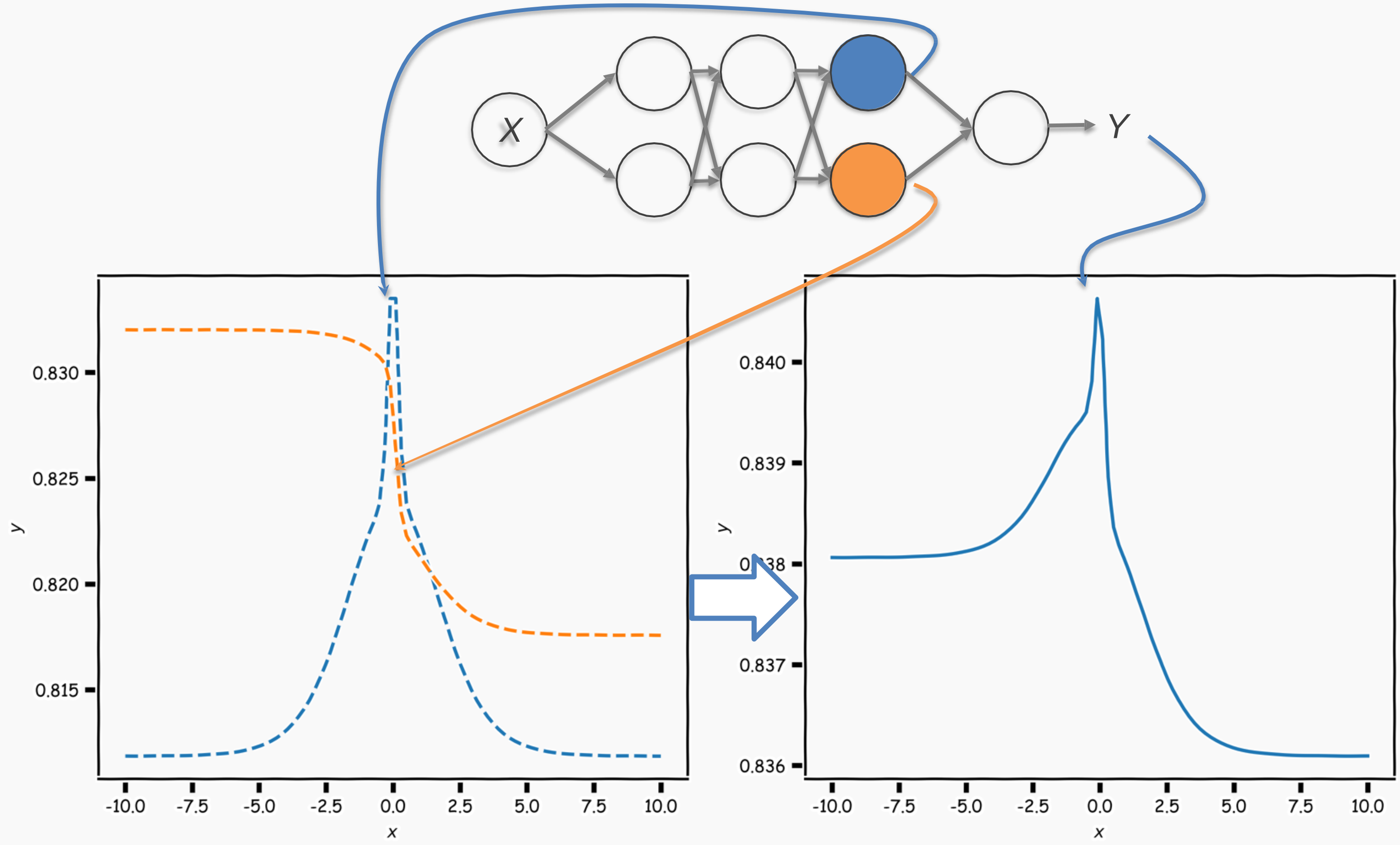
Different weights change the shape and position



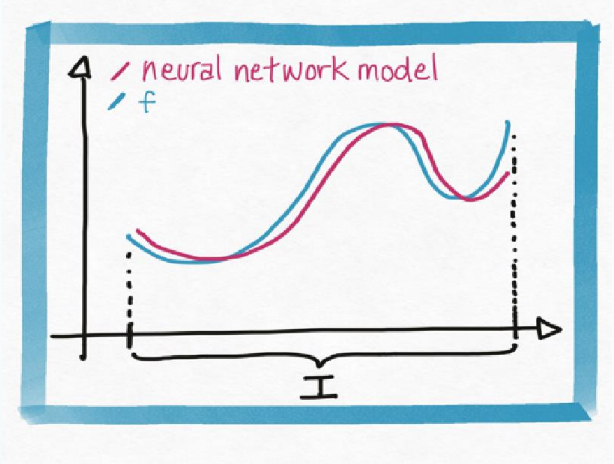
Neural networks can model *any* reasonable function



Adding layers allows us to model increasingly complex functions



Neural Networks as Universal Approximators



Theorem:

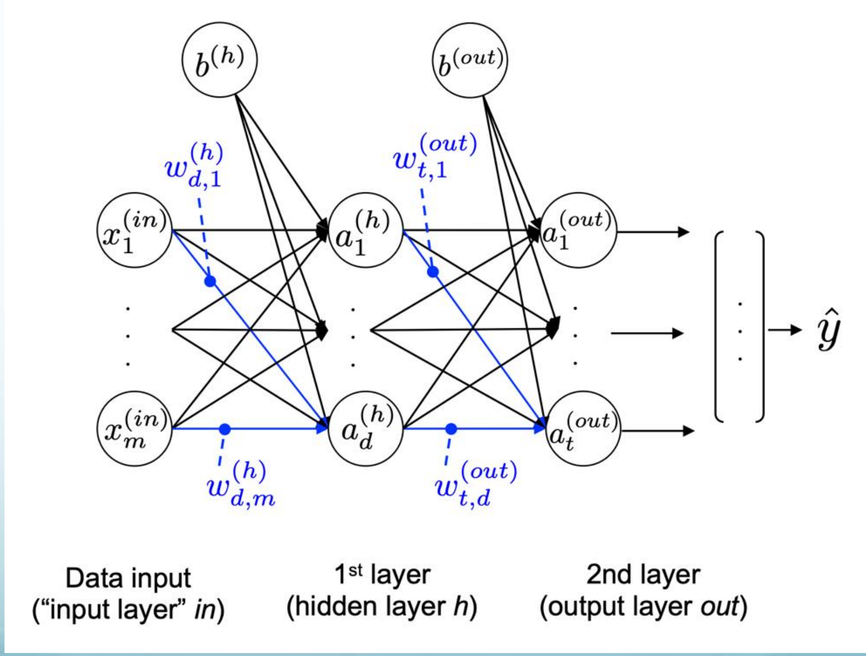
For any continuous function f defined on a bounded domain, we can find a neural network that approximates f with an arbitrary degree of accuracy.

One hidden layer is enough to represent an approximation of any function to an arbitrary degree of accuracy.

A neural network can approximate non-linear functions either for regression or classification.

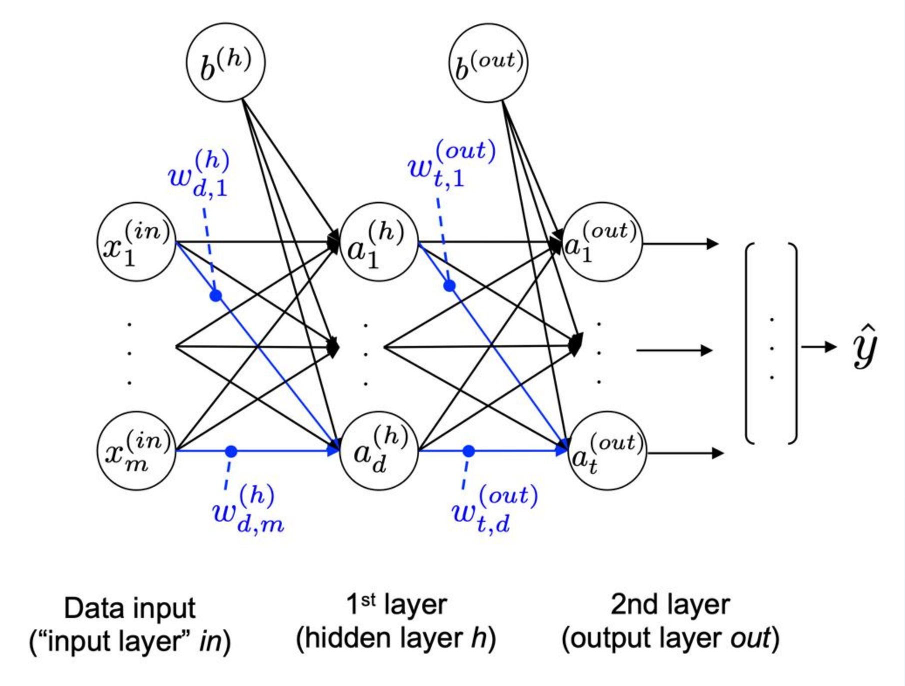
Multilayer Perceptron

- *A neural network is a combination of neurons such as logistic regression (or other types) units.*
- *A **multilayer perceptron (MLP)** is a fully connected network of neurons.*
- *An MLP is a multilayer **feedforward** NN because each layer is input to the next.*
- *A network with more than one hidden layer is a **deep NN.***

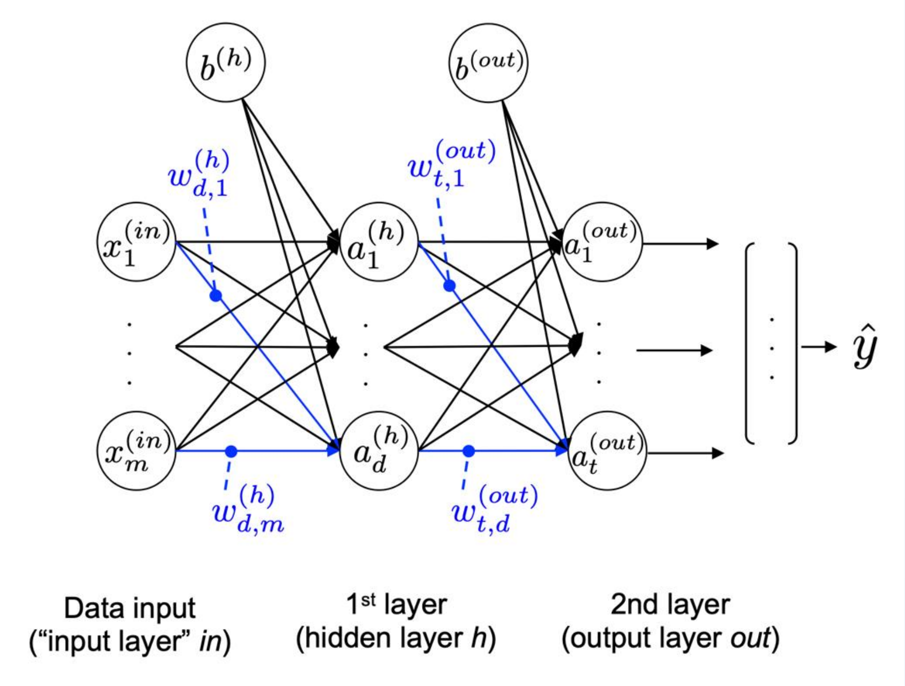


[Raschka et al 2022]

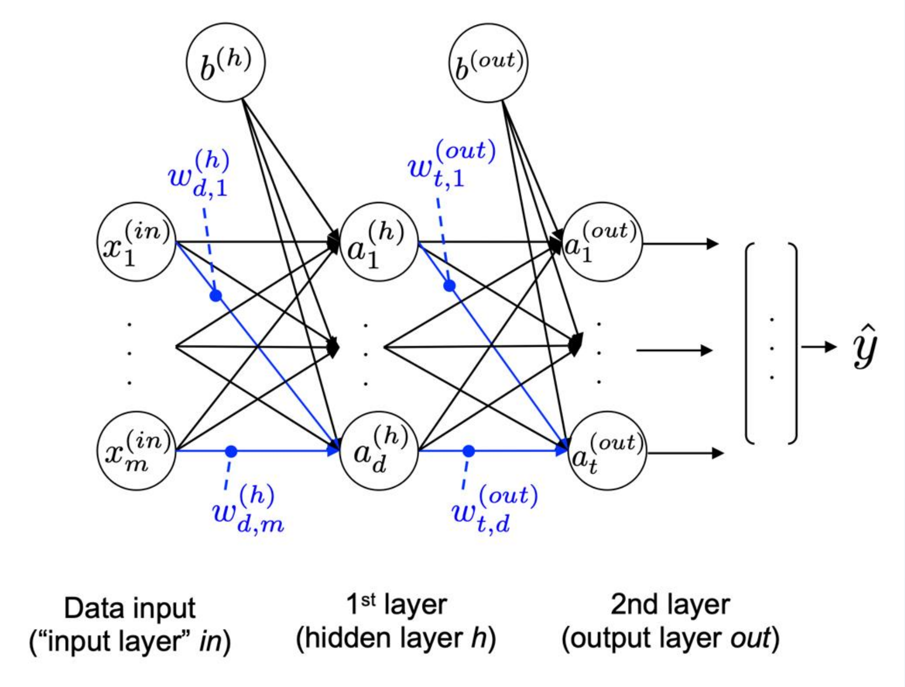
Multilayer Perceptron



Multilayer Perceptron



Multilayer Perceptron



Today's Learning Objectives

Students will be able to:

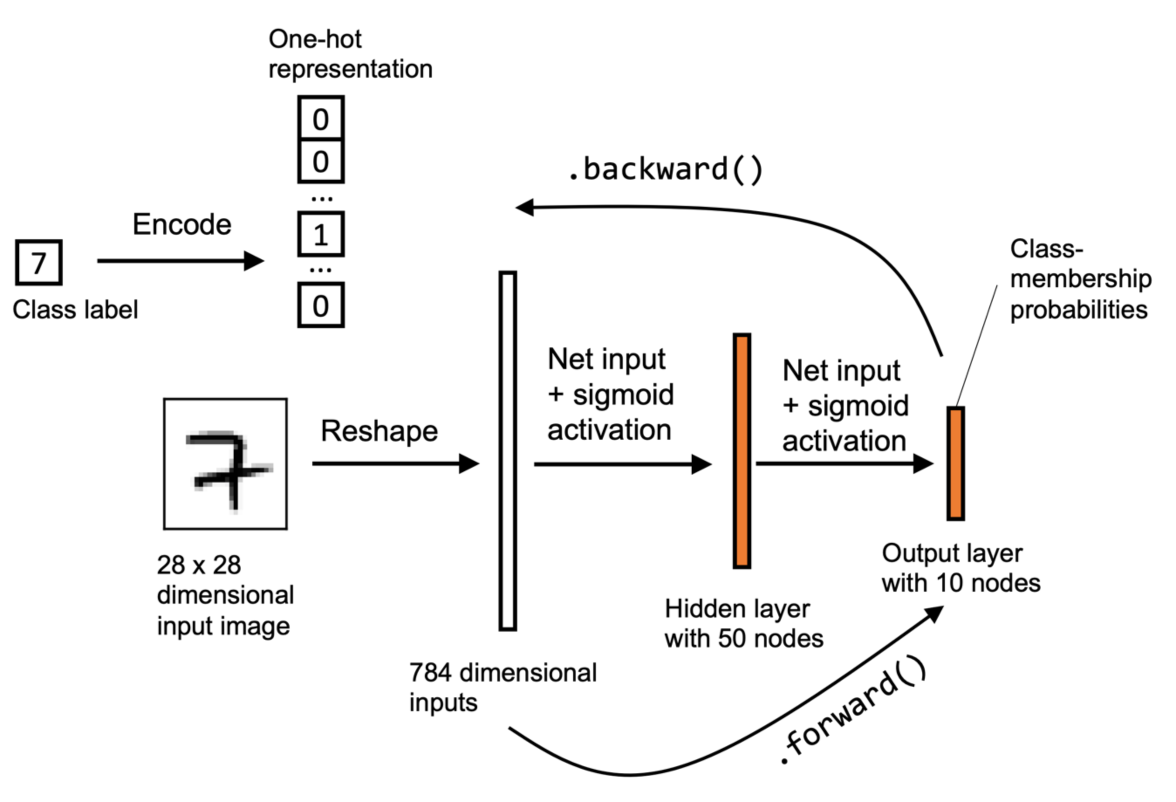
- ✔ Explain how Expectation Maximization (EM) is used for Clustering: Gaussian Mixture Models
- ✔ Apply the Forward and Backwards Algorithms to Hidden Markov Models (HMM)
- ✔ Describe what a neural network is
- Apply the forward algorithm on the MNIST NN

Modified National Institute of Standards and Technology (MNIST) Classification NN from Scratch

Hand-written digit recognition: MNIST data

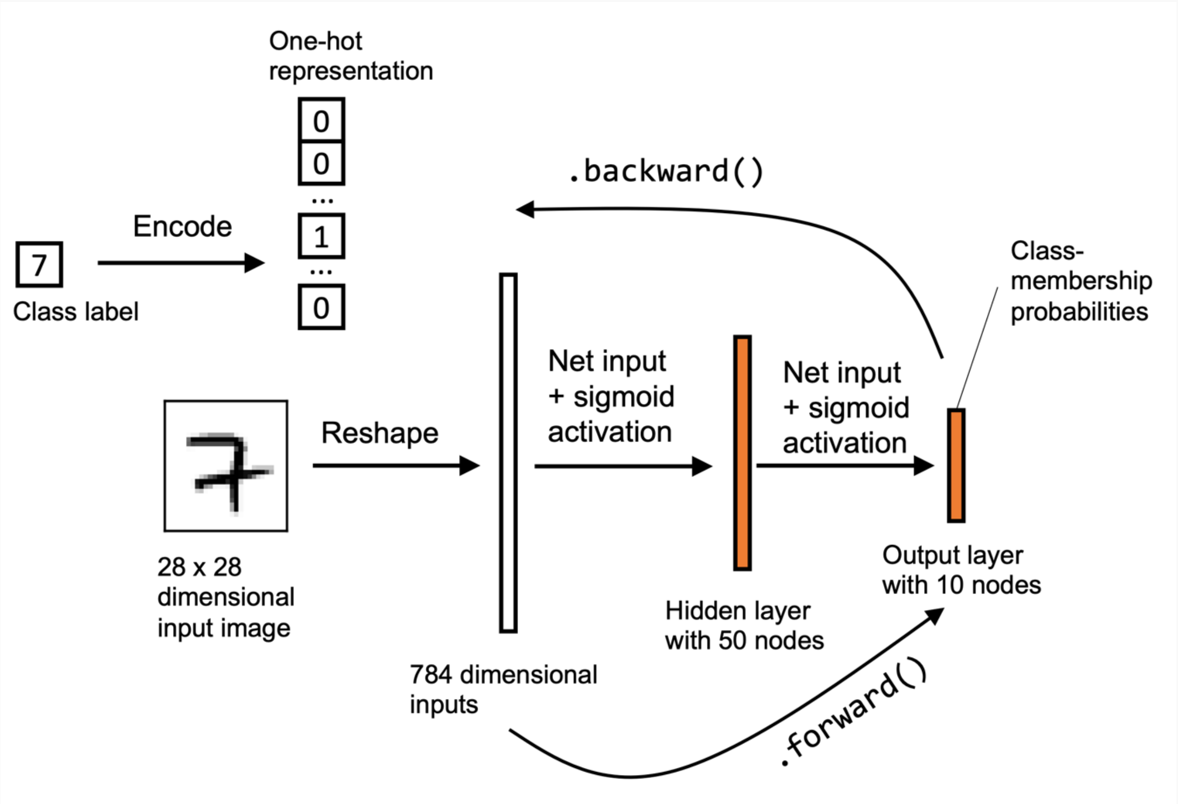
MNIST NN

Today we'll work on forward algorithm..
Next lecture backpropagation, gradient descent, training, and evaluating network performance



MNIST NN

Today we'll work on forward algorithm..
Next lecture backpropagation, gradient descent, training, and evaluating network performance



[Raschka et al 2022]

MNIST NN utility code

How would the following sample labels be one-hot encoded: [1, 0,2] ?

```
def sigmoid(z):
    return 1. / (1. + np.exp(-z))


def int_to_onehot(y, num_labels):

    ary = np.zeros((y.shape[0], num_labels))
    for i, val in enumerate(y):
        ary[i, val] = 1

    return ary
```


[Raschka et al 2022]

MNIST NN initialization

```
class NeuralNetMLP:

    def __init__(self, num_features, num_hidden, num_classes, random_seed=123):
        super().__init__()

        self.num_classes = num_classes

        # hidden
        rng = np.random.RandomState(random_seed)

        self.weight_h = rng.normal(
            loc=0.0, scale=0.1, size=(num_hidden, num_features))
        self.bias_h = np.zeros(num_hidden)

        # output
        self.weight_out = rng.normal(
            loc=0.0, scale=0.1, size=(num_classes, num_hidden))
        self.bias_out = np.zeros(num_classes)
```

[Raschka et al 2022]

MNIST NN Forward

How do we use the output for prediction?

```
def forward(self, x):
    # Hidden layer
    # input dim: [n_examples, n_features] dot [n_hidden, n_features].T
    # output dim: [n_examples, n_hidden]
    z_h = np.dot(x, self.weight_h.T) + self.bias_h
    a_h = sigmoid(z_h)

    # Output layer
    # input dim: [n_examples, n_hidden] dot [n_classes, n_hidden].T
    # output dim: [n_examples, n_classes]
    z_out = np.dot(a_h, self.weight_out.T) + self.bias_out
    a_out = sigmoid(z_out)
    return a_h, a_out
```


[Raschka et al 2022]

Backpropagation Algorithm

We'll start here next time, but a question to consider….

How was the backward forward algorithm we already considered for the casino an example of this?

Today's Learning Objectives

Students will be able to:

- ✓ Explain how Expectation Maximization (EM) is used for Clustering: Gaussian Mixture Models
- ✓ Apply the Forward and Backwards Algorithms to Hidden Markov Models (HMM)
- ✓ Describe what a neural network is
- ✓ Apply the forward algorithm on the MNIST NN

Citations:.

Shah. C. (2020) A hands-on introduction to data science. Cambridge University Press.

Sebastian **Raschka**, Yuxi (Hayden) **Liu**, and Vahid Mirjalili. **Machine Learning** with PyTorch and Scikit-Learn. Packt Publishing, **2022**.

Baharan Mirzasoleiman, UCLA CS M148 Winter 2024 Lecture 13 Notes

Some slides from https://www.molgen.mpg.de/3379148/Hidden_Markov_Models_vin.pdf

Thank You