outraggeously
AMBITIOUS

Module 6: Deep Learning

Duke
PRATT SCHOOL *of* ENGINEERING

Module 6 Objectives:

At the conclusion of this module, you should be able to:

1) Describe the intuition and mathematical principles behind deep learning
2) Identify common applications of deep learning for computer vision and NLP
3) Explain the strengths and challenges of deep learning relative to other forms of ML

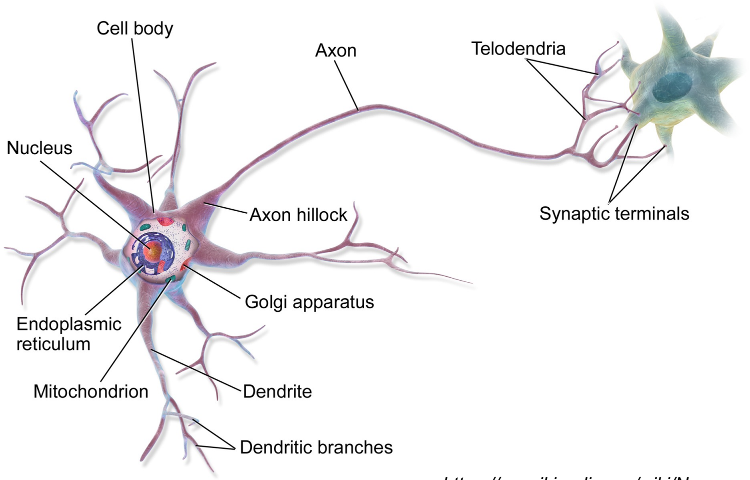
outrageously
AMBITIOUS

Introduction to Deep Learning

Duke
PRATT SCHOOL *of* ENGINEERING

What is a Neuron?

- In 1943, the neurophysiologist Warren McCulloch and mathematician Walter Pitts introduced a computational model for how neurons work together to perform complex computations
- Multiple signals arrive at dendrites, they are added together in the cell body and if the accumulated signal exceeds a threshold, the neuron is activated and passes on an output signal



https://en.wikipedia.org/wiki/Neuron

Neural Networks

- Each neuron may be connected to thousands of other neurons, enabling complex calculations
- Likewise, the combination of multiple artificial neurons in layers enables a neural network to approximate complex functions
- A neural network with many layers is called a **deep neural network**

History of Neural Networks

1940s 1950s 1960s 1970s 1980s 1990s 2000s 2010s 2020

Early AI research

1943: McCulloch & Pitts propose simple model of neuron

1957: Rosenblatt publishes the perceptron – first artificial neuron

1958: Stanford researchers Widrow & Hoff develop the first successful neural network

”AI Winter” 1

1982: CalTech researcher Hopfield’s bi-directional neural net revives research efforts

1986: Three independent groups develop back-propagation method for training neural networks

”AI Winter” 2

Deep learning boom

2009: ImageNet competition created by Fei-Fei Li of Stanford

2012: AlexNet wins ImageNet, leading to widespread use of GPUs and CNN architectures

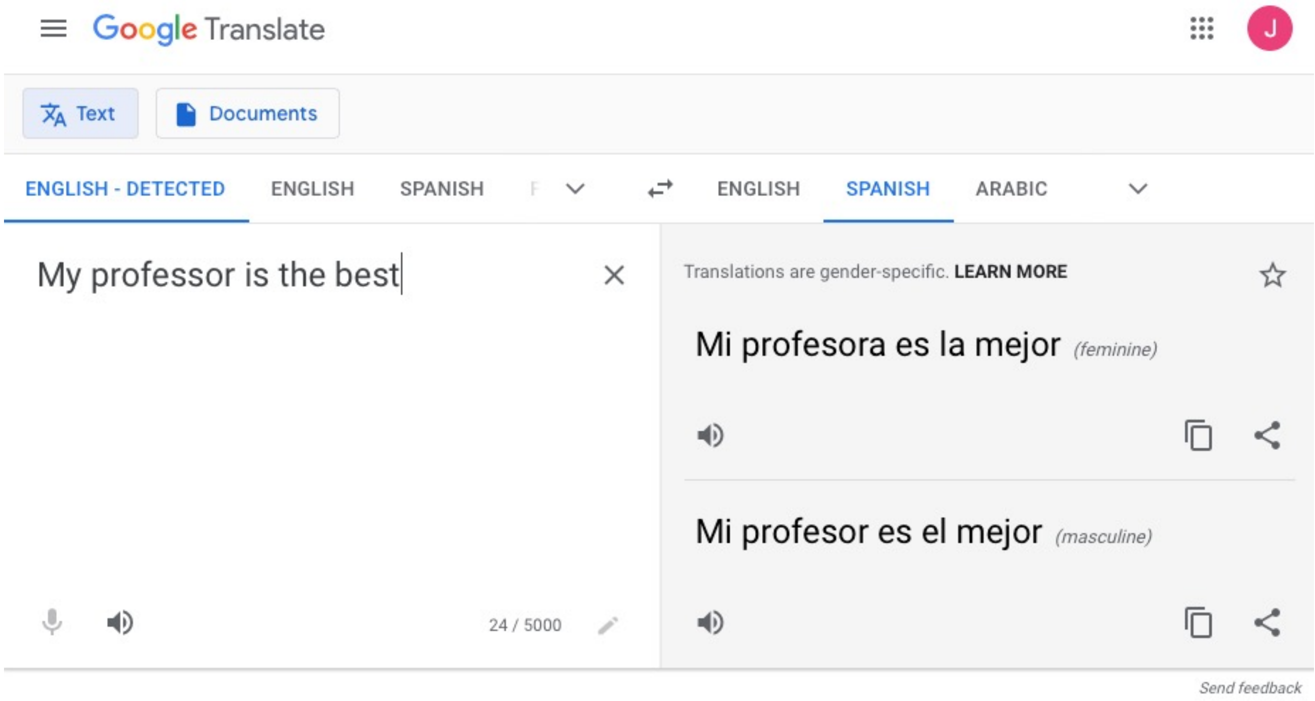
2015: Deep ResNet architecture greatly advances performance

Enablers of the deep learning boom

1. The amount of data available for training has grown exponentially
2. Efforts have been made to label vast amounts of data for training
3. Computational power has increased tremendously, allowing for much deeper neural nets
4. Major algorithmic advances have overcome limitations of neural nets

Applications of deep learning

Applications of deep learning



Applications of deep learning

Applications of deep learning

Applications of deep learning

Deep learning excels in applications which have:

1. Vast amounts of training data
2. Very large number of features – e.g. unstructured data
3. Complex relationships between features and the target
4. Low concern for explainability

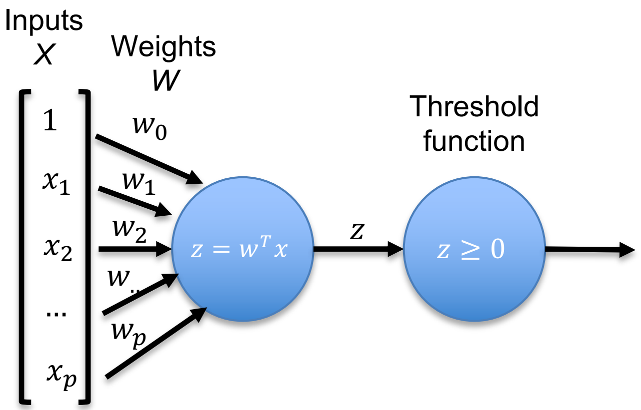
outrageously
AMBITIOUS

Artificial Neurons

Duke
PRATT SCHOOL of
ENGINEERING

The Perceptron

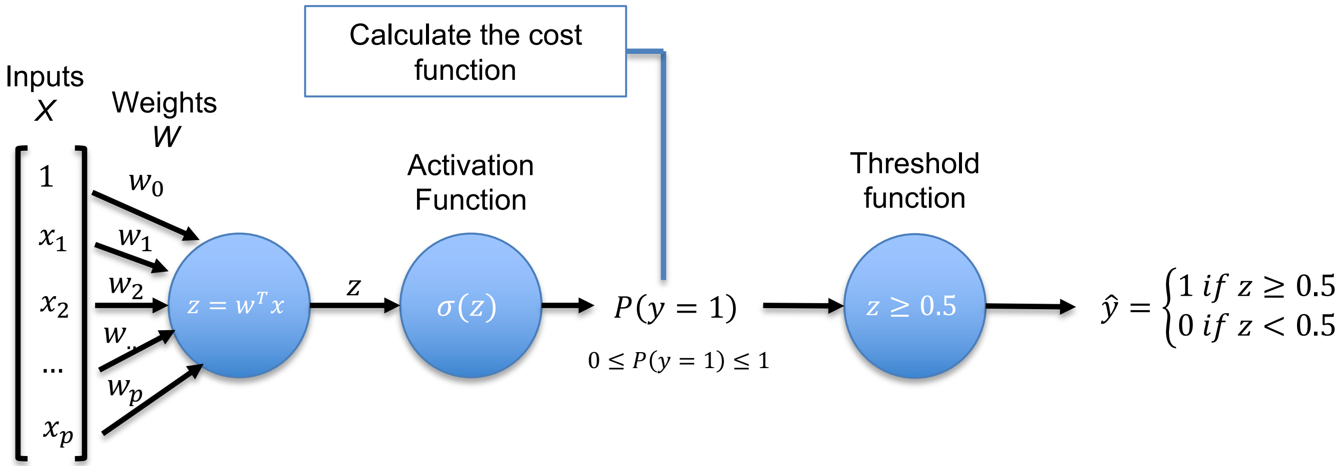
- In the perceptron, a set of weight coefficients is multiplied by the input features $x_0 \ldots x_m$ and the results are summed together
- The sum is then passed through a **threshold function** and if it exceeds a threshold (0), the neuron is activated



$$\hat{y} = \begin{cases} 1 \; if \; z \geq 0 \\ -1 \; if \; z < 0 \end{cases}$$

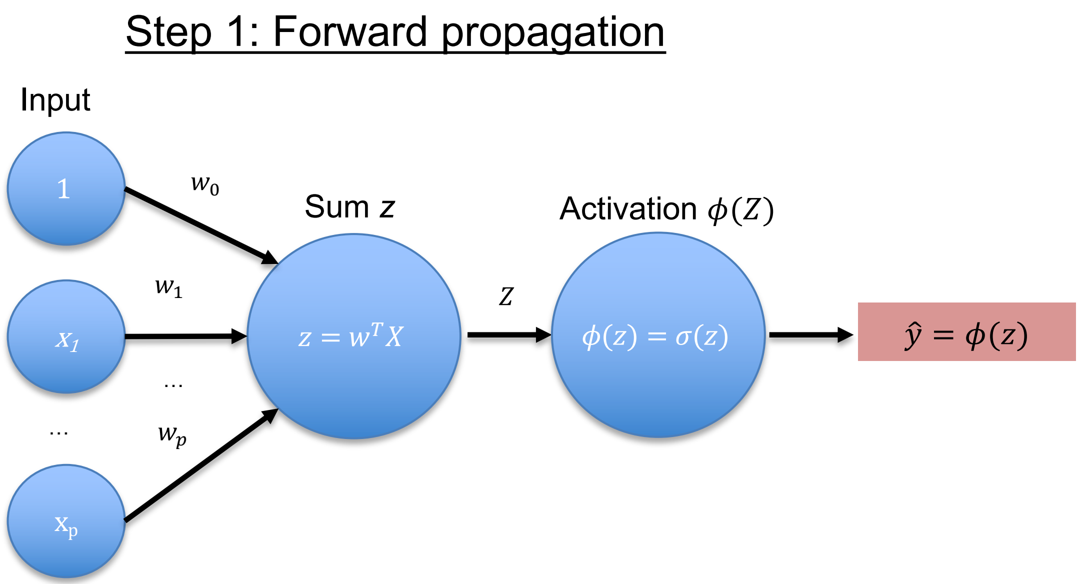
Logistic Regression

- Similar to the Perceptron, but we've added in an **activation function** (sigmoid function)
- We use the output of the activation function to calculate the cost
- Our objective is to find the weight values that minimize the cost



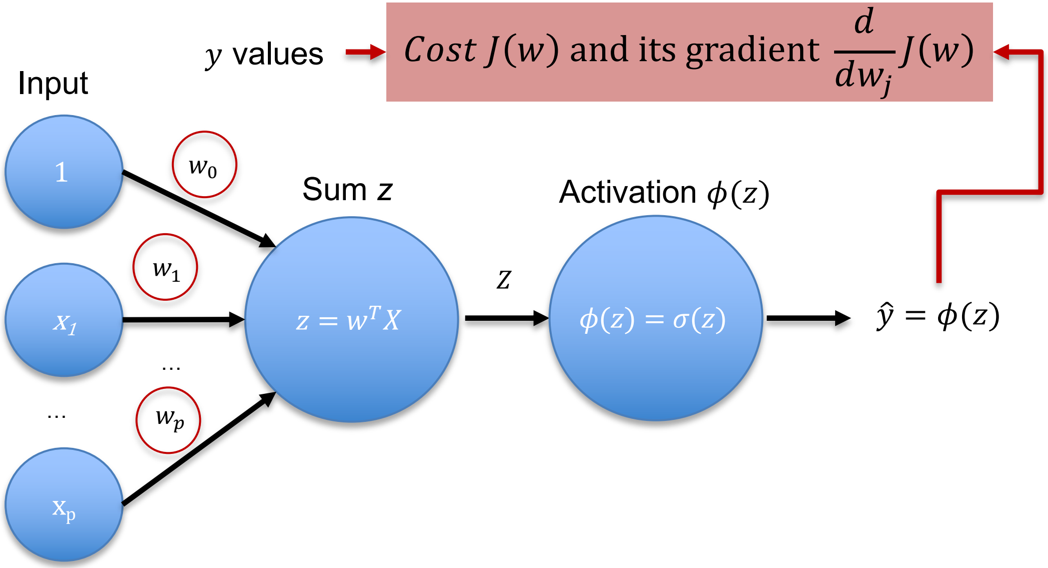
Training a neuron

Let's start by walking through the process for a single neuron with a sigmoid activation function (logistic regression) using SGD



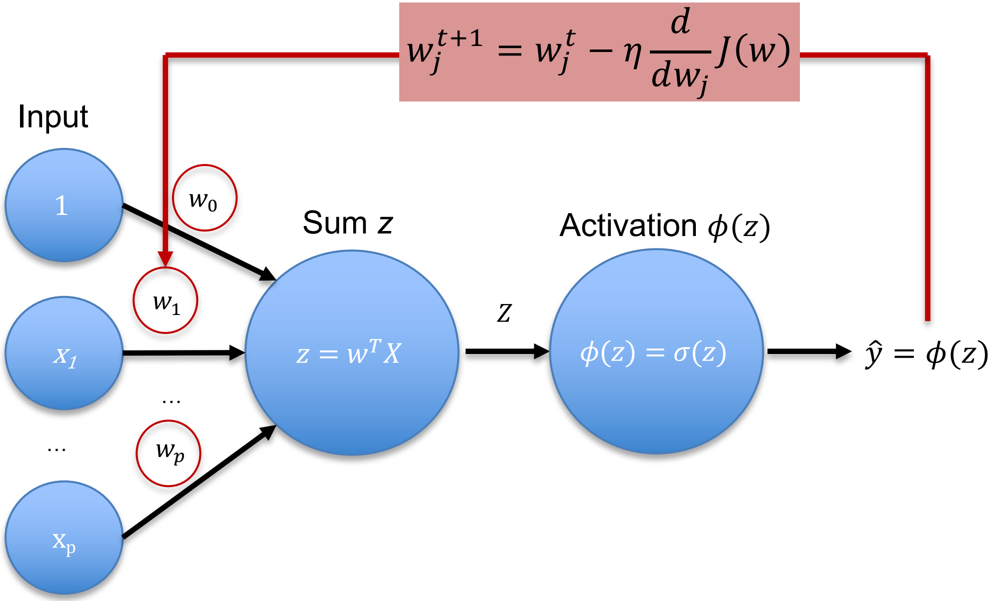
Training a neuron

Step 2: Calculate the gradient of the cost function with respect to each weight



Training a neuron

Step 3: Update each weight using gradient descent and repeat

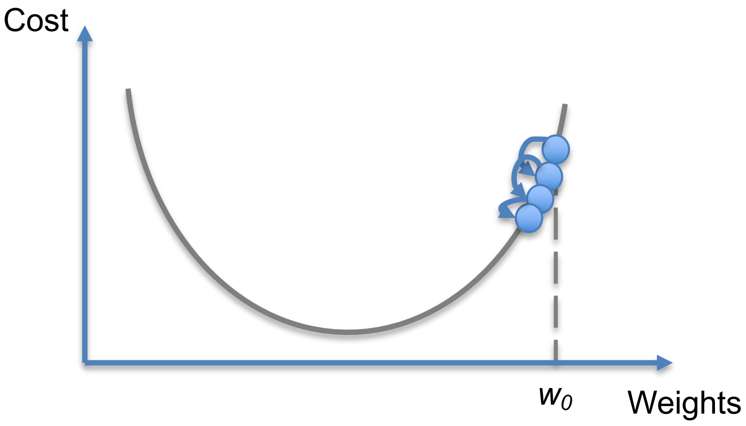


Learning rate

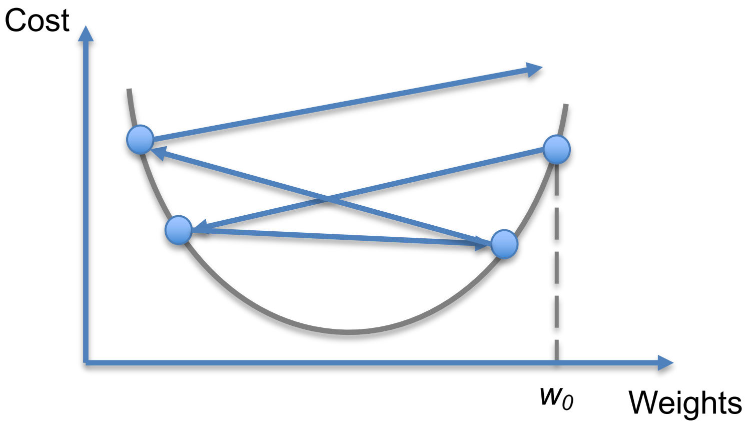
$$w_{t+1} = w_t - \textcolor{red}{learning\ rate\ \eta} * gradient\ of\ cost\ fn$$

- If too small, algorithm may take a very long time to converge
- If too large, the gradient will bounce around and may even diverge

Learning rate too small



Learning rate too large



Methods of gradient descent

	Stochastic (SGD)	Batch	Mini-batch
Methodology	Loop through observations using one at a time to iteratively calculate gradient and update weights	Calculate the gradient and weights update based on all observations in the training set summed together for each iteration	Divide training data into subsets of N (e.g. 8,32) observations and perform batch gradient descent on each subset at a time
Pros	Works well for large datasets and online learning	Can use vectorized operations	Works well for large datasets and uses vectorized operations
Cons	Cannot use vectorized linear algebra operations	May be impossible for large datasets	Not as good as SGD for online learning

outrageously
AMBITIOUS

From Neurons to Neural Networks

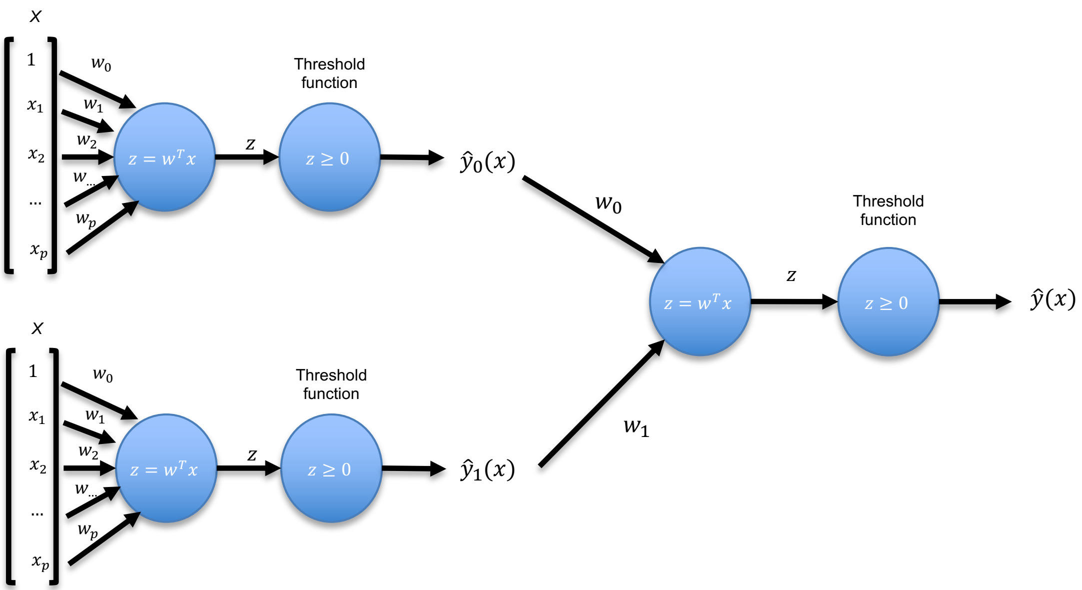
Duke
PRATT SCHOOL *of*
ENGINEERING

One neuron is good, more is better!

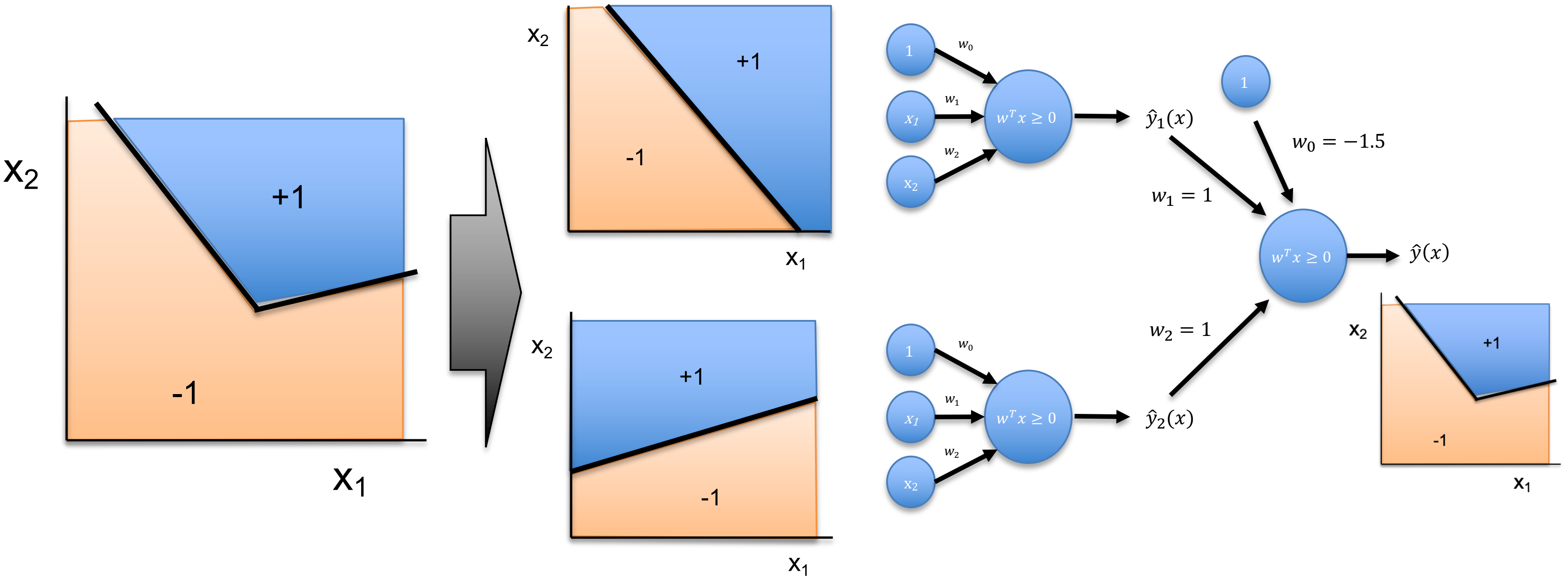
- Power of artificial neurons was limited because they could only handle problems with linear decision boundaries
- Researchers knew adding more neurons to form a network would allow for much more complex calculations
- However, there was no good way to train multi-layer neural networks until the **backpropagation** method was popularized

Multilayer perceptron (MLP)

Since a single neuron can only represent a linear decision boundary, what if we stacked multiple neurons together in layers?

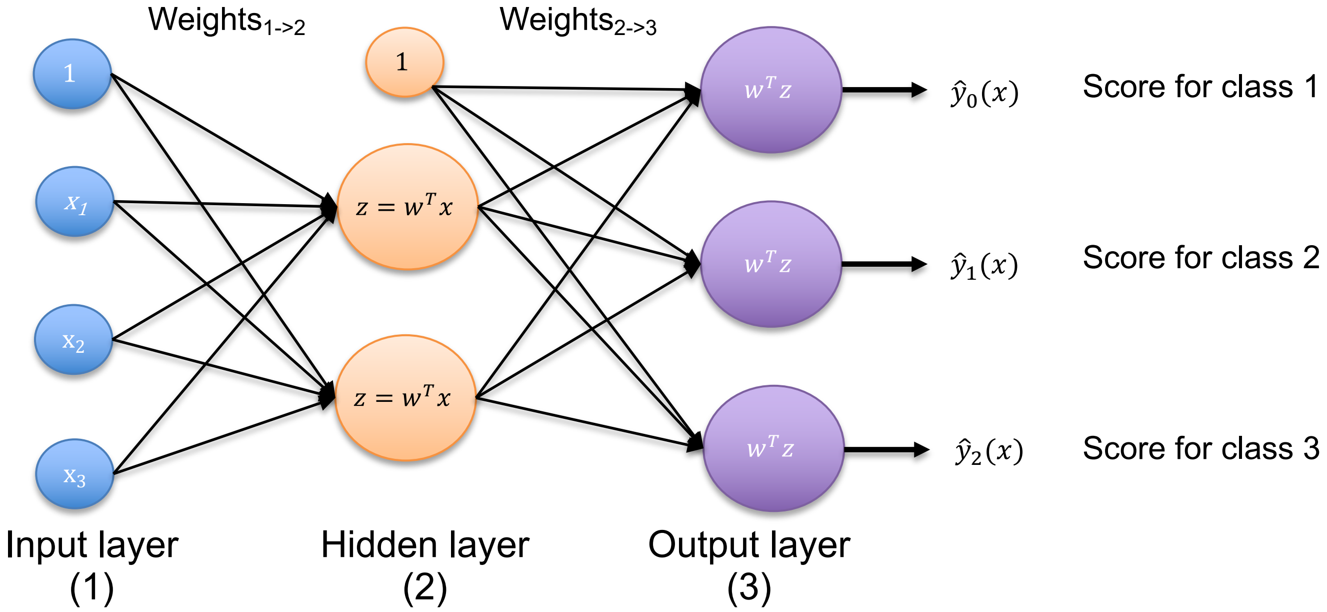


Multilayer perceptron (MLP)



MLPs for multiclass problems

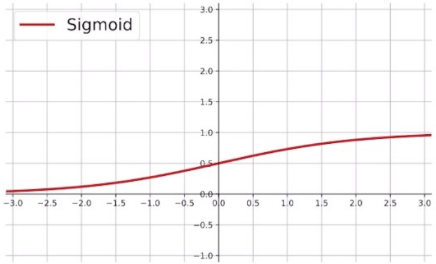
What if we have more than 2 possible output classes? We use multiple units in the output layer and assign the label to the unit with the highest score



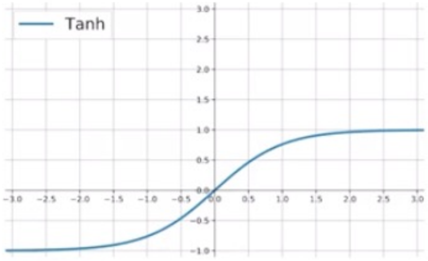
From MLPs to neural networks

- Rather than using a perceptron with a simple threshold function for each unit, we can choose to use units with an activation function
- The use of non-linear activation functions in each layer enables us to better model non-linear relationships

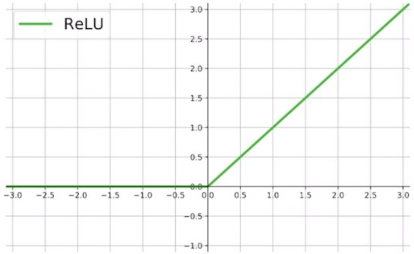
<u>Sigmoid</u>



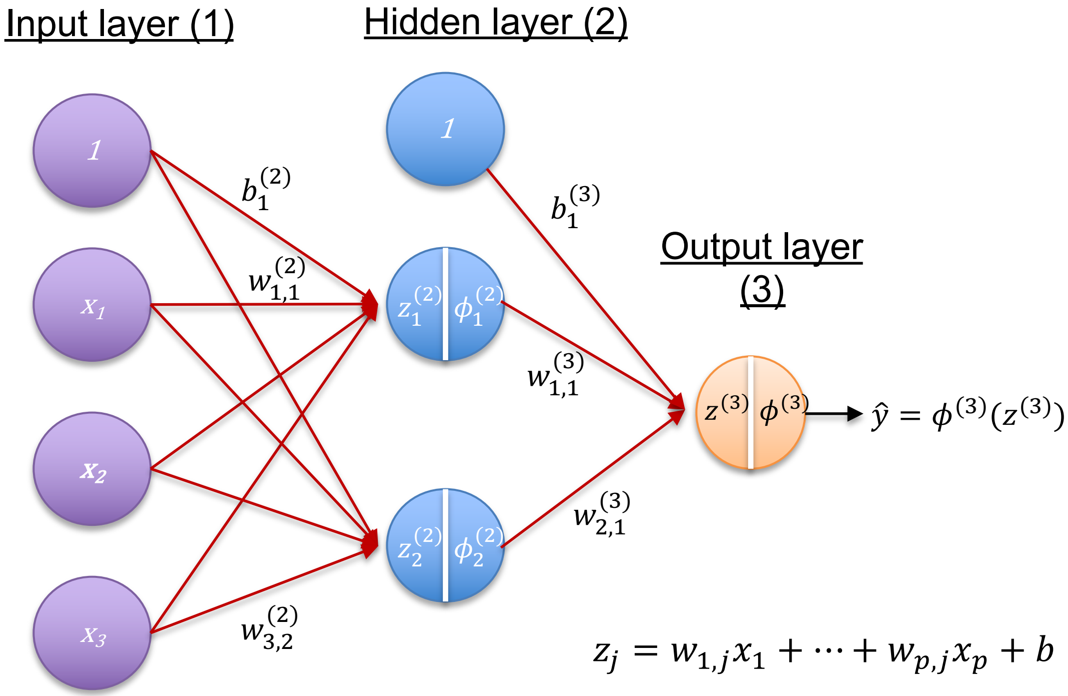
<u>Hyperbolic tangent (Tanh)</u>



<u>ReLU</u>



Neural network architecture



$$z_j = w_{1,j}x_1 + \cdots + w_{p,j}x_p + b$$

$$\phi(z_j) = activation\ function$$

outrageously AMBITIOUS

Training Neural Networks

Duke

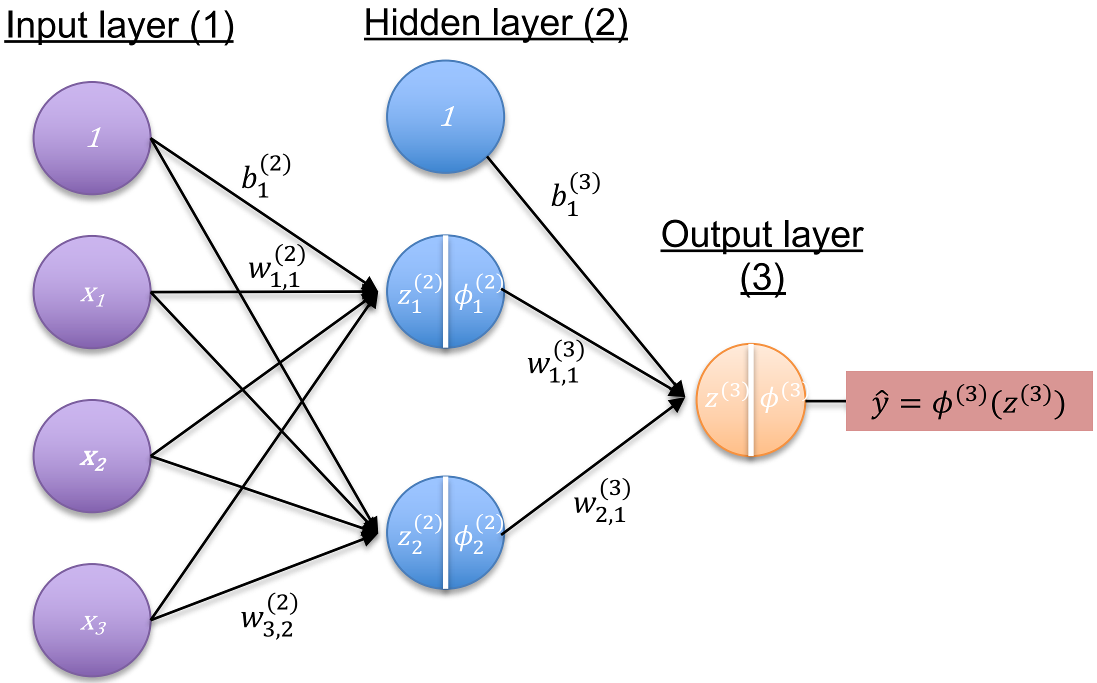
PRATT SCHOOL *of* ENGINEERING

Training a neural network

- In a neural network we have multiple layers of weights to update
- We work in reverse from right to left, distributing the total output error among each layer
- We can then calculate the gradient of the layer error with respect to each weight and perform gradient descent on the weights
- This process is called **backpropagation**

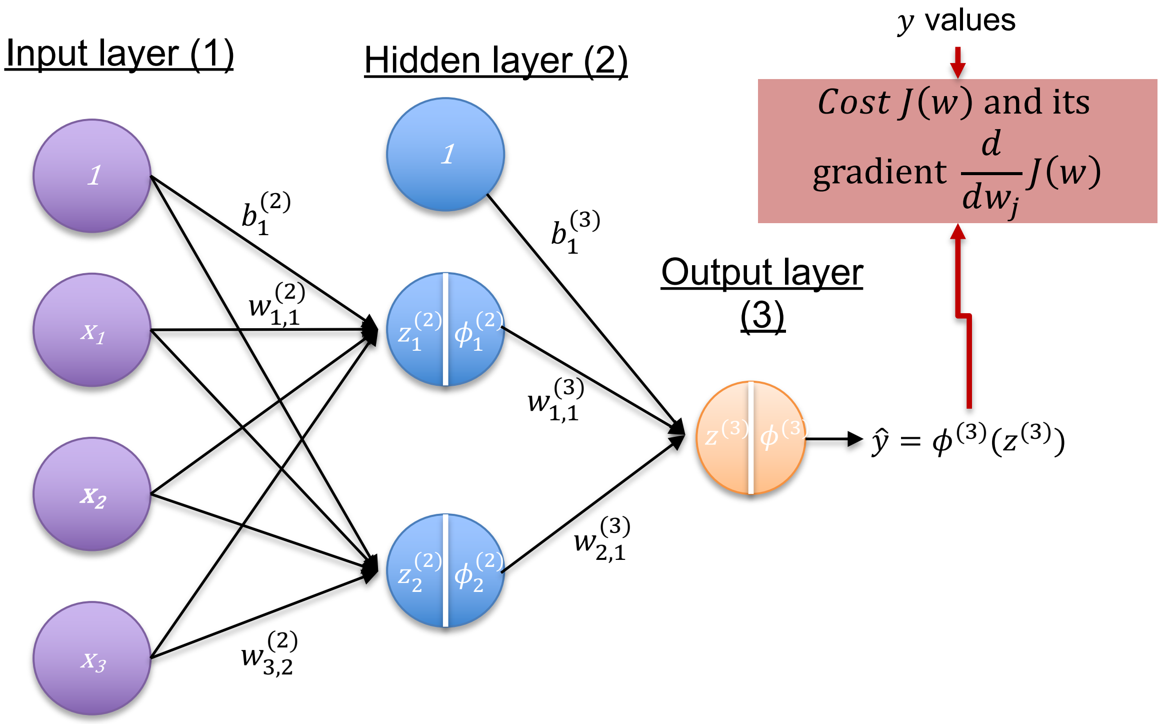
Training a neural network

Step 1: Forward propagation



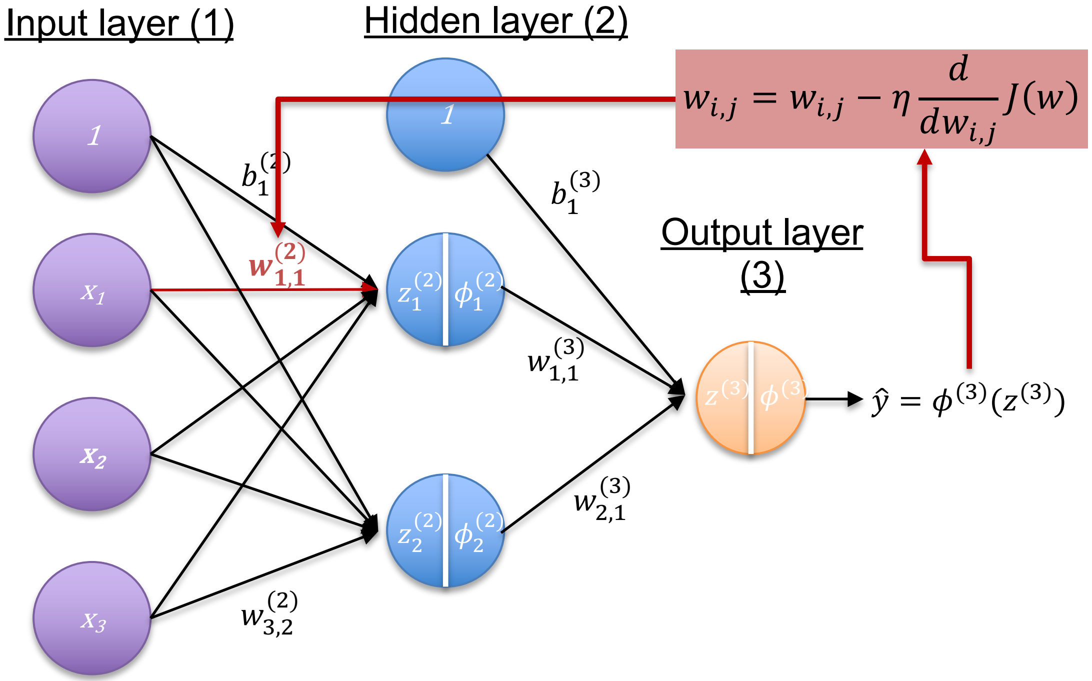
Training a neural network

Step 2: Calculate gradient of the cost with respect to each weight



Training a neural network

Step 3: Use backpropagation to update the weight values

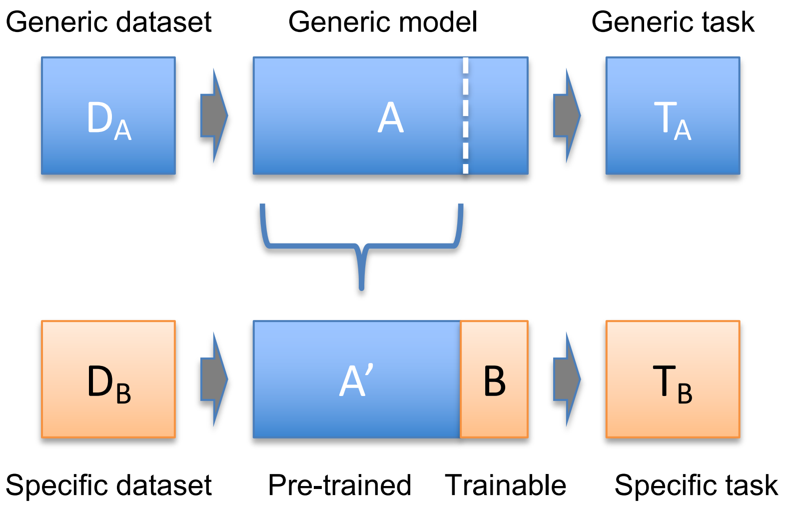


Neural network design

- When setting up a NN, we have many decisions:
  - How many layers?
  - How many units in each layer?
  - Activation function for each layer?
  - Add regularization?
  - Batch vs. mini-batch vs SGD?
  - Choice of learning rate?
- In practice, we often:
  1. Use the “stretch pants” approach and apply techniques to reduce overfitting
  2. Use a neural network that someone has set up and pre-trained

Transfer learning

- In transfer learning, we start from a pre-built, pre-trained neural net trained on a relevant task (e.g. classifying images)
- We then add our own final layers which we train on our specific task
- We benefit from the (significant) earlier training without having to do it



outraged**ously**

outrageously
AMBITIOUS

Computer Vision

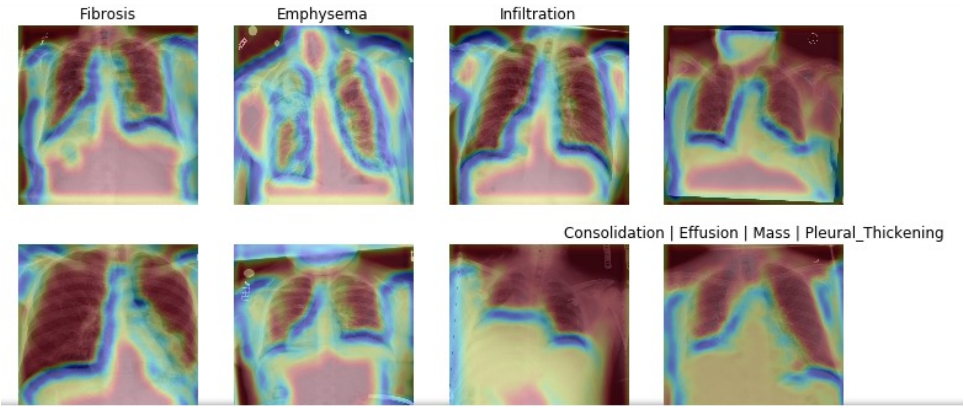
Duke
PRATT SCHOOL *of* ENGINEERING

Computer vision tasks

- Image classification
- Object detection
- Semantic segmentation
- Image generation

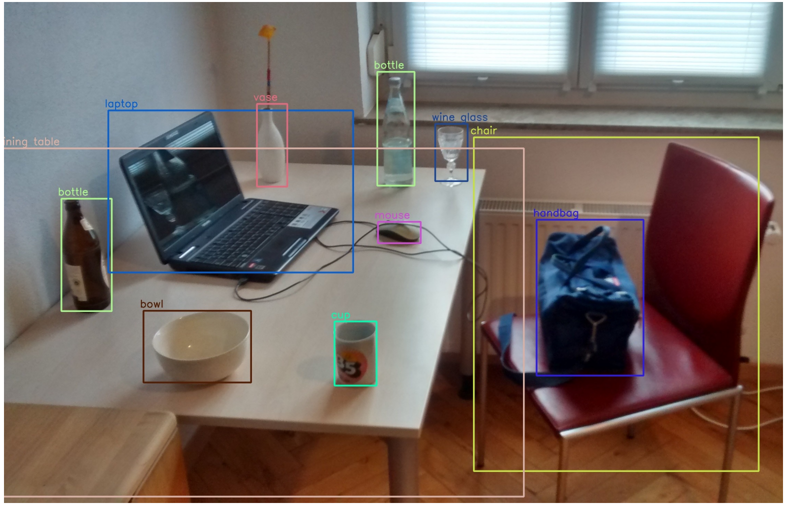
Computer vision tasks

- **Image classification**
- Object detection
- Semantic segmentation
- Image generation



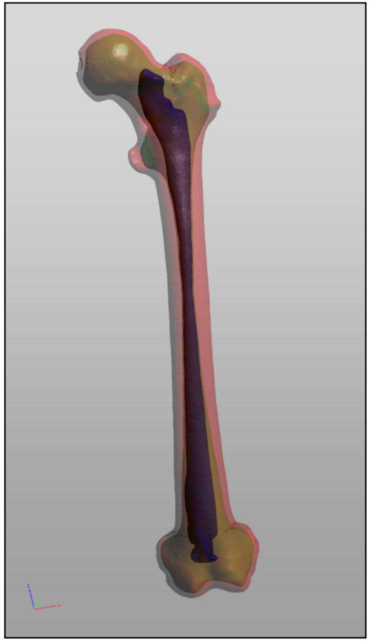
Computer vision tasks

- Image classification
- **Object detection**
- Semantic segmentation
- Image generation



Computer vision tasks

- Image classification
- Object detection
- **Semantic segmentation**
- Image generation

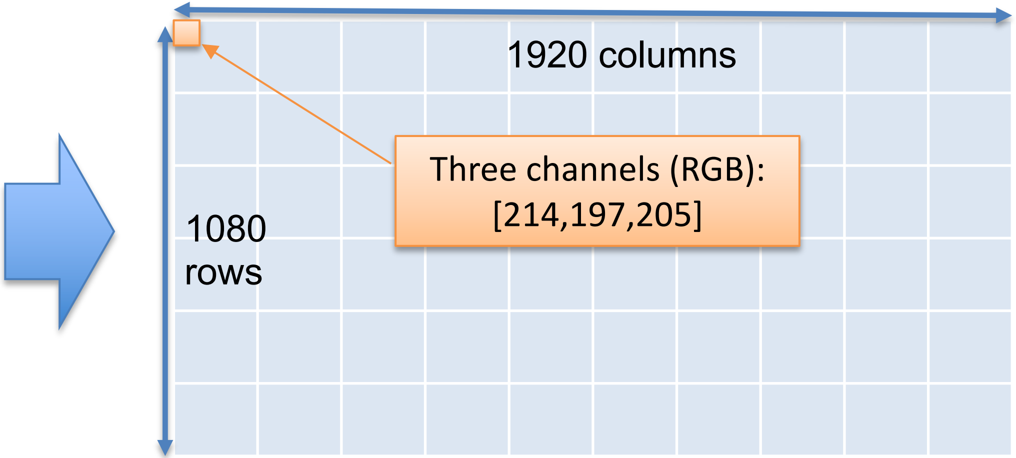


Computer vision tasks

- Image classification
- Object detection
- Semantic segmentation
- **Image generation**

Working with images

- Color of each pixel can be represented as a number
- The most common color encoding is RGB
  - Red, green, blue in each pixel represented on scale from 0 to 255
- We can use the pixel values as the input features to train models



Convolutional neural networks

- When working with many features, the number of weights can become extremely large
- **Convolutional neural networks** (CNNs) are neural networks which utilize two additional types of layers:
  - **Convolutional** layers – act as filters to learn patterns in data
  - **Pooling** layers – reduce dimensions of data

Convolutional layers

- Convolution layers utilize a sliding filter containing weights and apply it across the entire input set
- Each node is connected to a limited subset of the nodes in the layer before
- The filters contain shared weights which define the connections between layers
- During training, each filter “learns” to recognize certain patterns

Input image

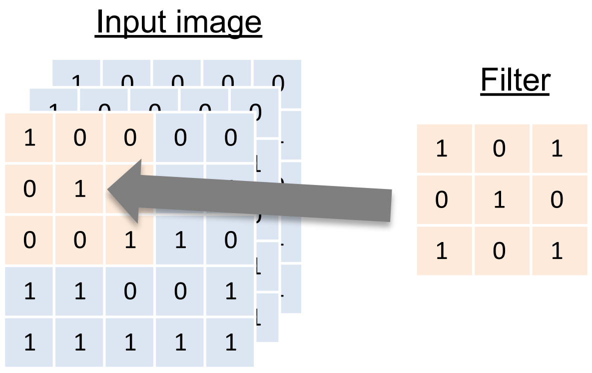
1	0	0	0	0
0	1	1	0	1
0	0	1	1	0
1	1	0	0	1
1	1	1	1	1

Filter

1	0	1
0	1	0
1	0	1

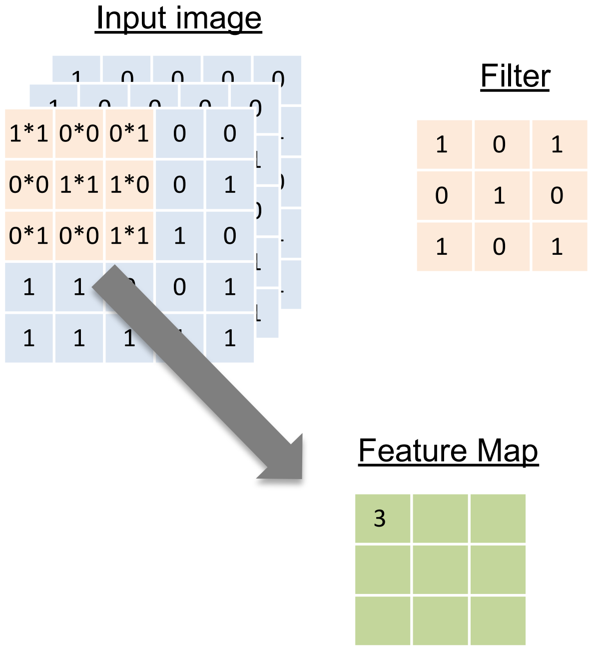
Convolutional layers

- Convolution layers utilize a sliding filter containing weights and apply it across the entire input set
- Each node is connected to a limited subset of the nodes in the layer before
- The filters contain shared weights which define the connections between layers
- During training, each filter “learns” to recognize certain patterns



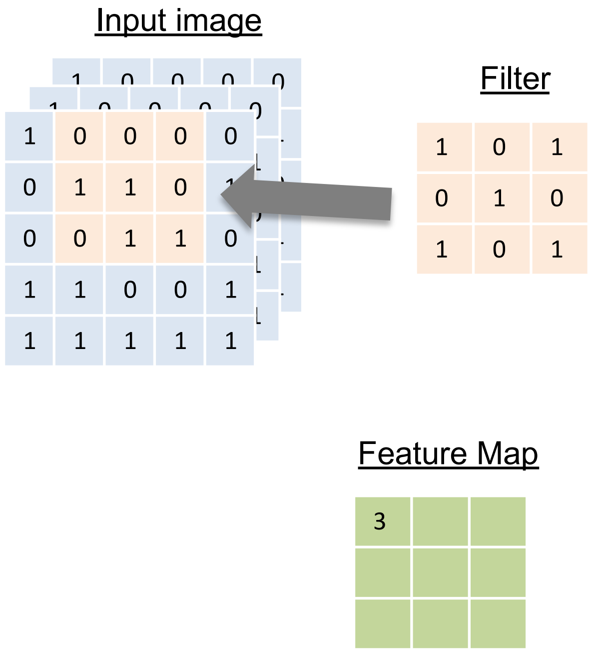
Convolutional layers

- Convolution layers utilize a sliding filter containing weights and apply it across the entire input set
- Each node is connected to a limited subset of the nodes in the layer before
- The filters contain shared weights which define the connections between layers
- During training, each filter “learns” to recognize certain patterns



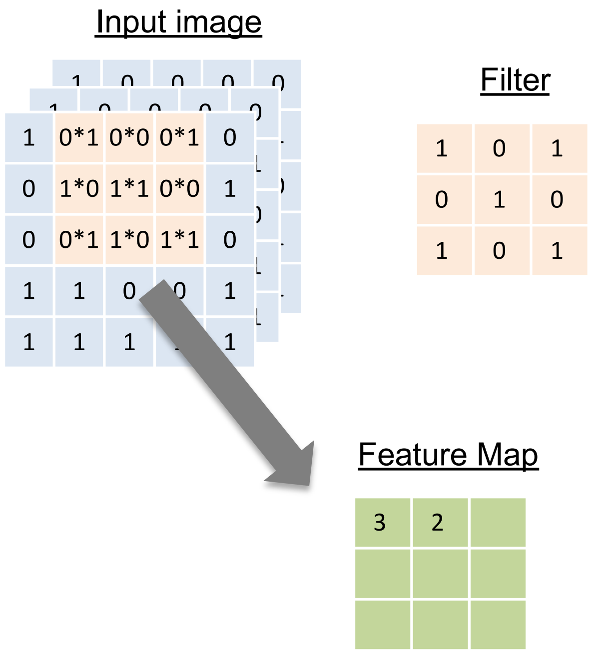
Convolutional layers

- Convolution layers utilize a sliding filter containing weights and apply it across the entire input set
- Each node is connected to a limited subset of the nodes in the layer before
- The filters contain shared weights which define the connections between layers
- During training, each filter “learns” to recognize certain patterns

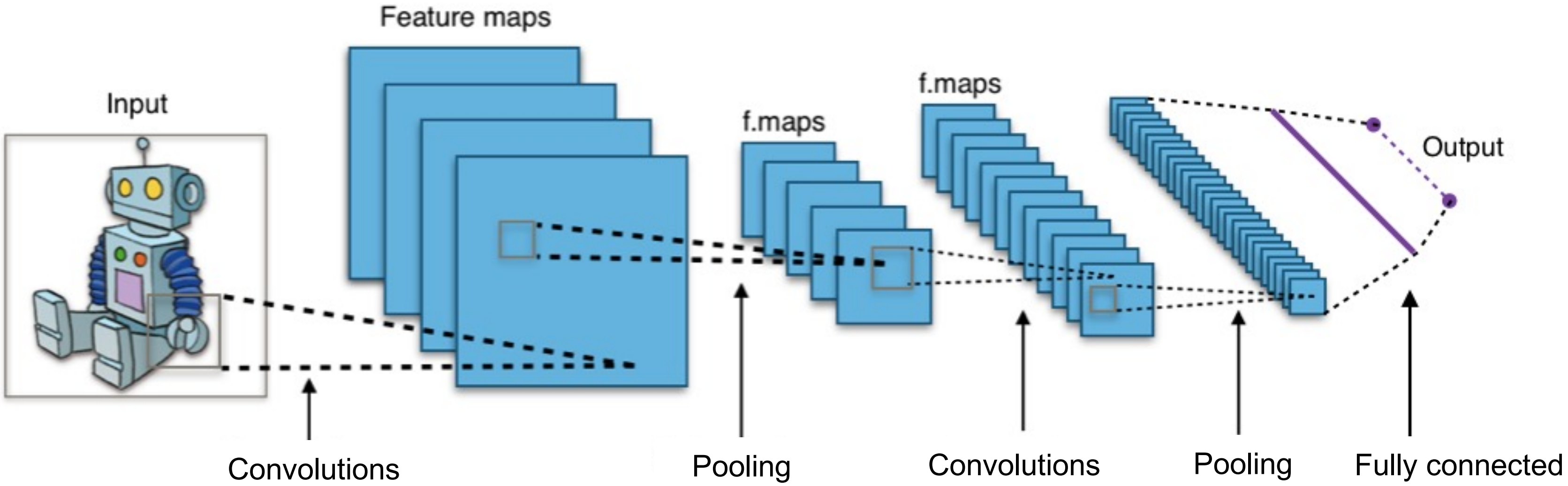


Convolutional layers

- Convolution layers utilize a sliding filter containing weights and apply it across the entire input set
- Each node is connected to a limited subset of the nodes in the layer before
- The filters contain shared weights which define the connections between layers
- During training, each filter “learns” to recognize certain patterns

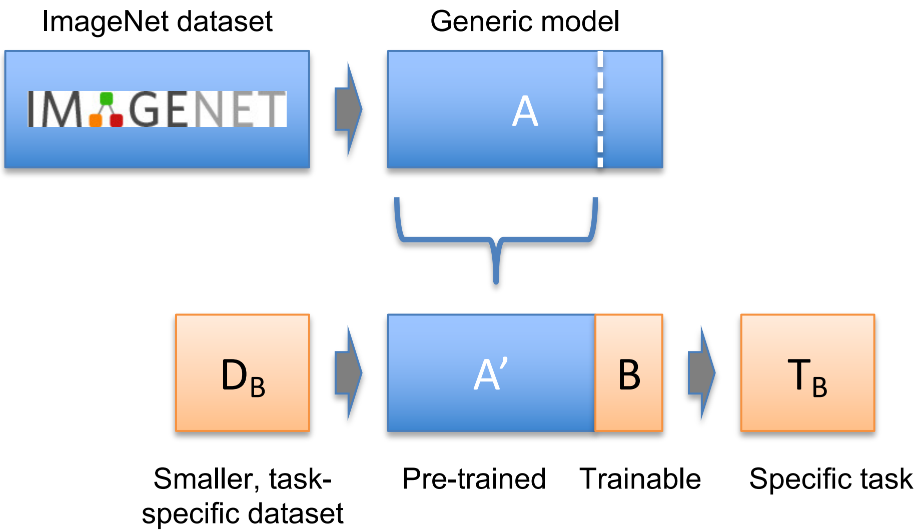


Convolutional neural network



Transfer learning via ImageNet

ImageNet is a database of over 14 million images organized into ~20,000 categories



outrageously
AMBITIOUS

Natural Language Processing

Duke
PRATT SCHOOL of
ENGINEERING

Common NLP tasks

- Text classification
- Sentiment analysis
- Search
- Machine translation
- Text generation

Common NLP tasks

- **Text classification**
- Sentiment analysis
- Search
- Machine translation
- Text generation

Common NLP tasks

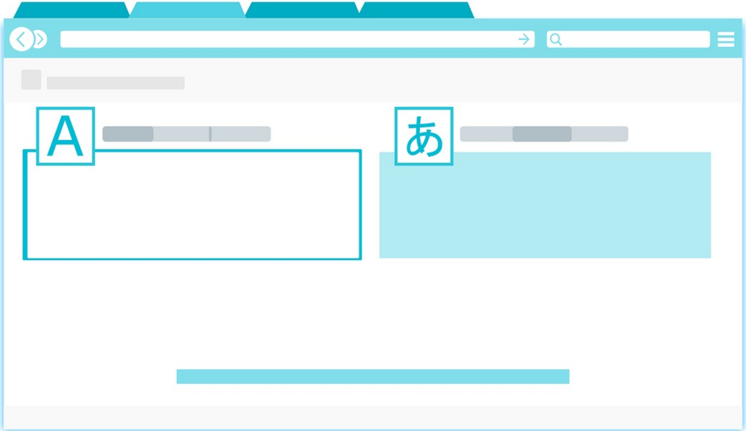
- Text classification
- **Sentiment analysis**
- Search
- Machine translation
- Text generation

Common NLP tasks

- Text classification
- Sentiment analysis
- **Search**
- Machine translation
- Text generation

Common NLP tasks

- Text classification
- Sentiment analysis
- Search
- **Machine translation**
- Text generation



Common NLP tasks

- Text classification
- Sentiment analysis
- Search
- Machine translation
- **Text generation**

Working with text

- Text must be encoded into features to use in models
- There are multiple ways to represent text for modeling:
  - Vocabulary – “Bag of Words “
  - Word / document embeddings
  - Attention (Transformers)

Bag of words

Review 1:

"The movie was the best"

Review 2:

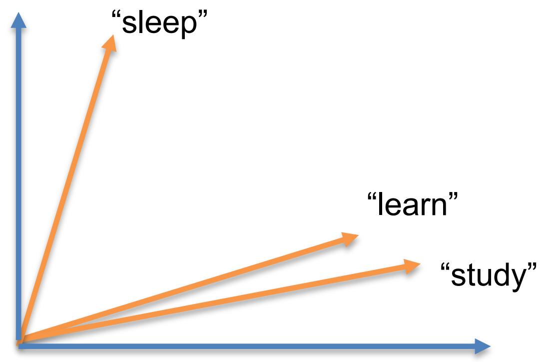
"The movie was long and boring"

Vocabulary:
["The", "movie", "was", "best", "long", "and", "boring"]

	"The"	"movie"	"was"	"best"	"long"	"and"	"boring"
Review 1	2	1	1	1	0	0	0
Review 2	1	1	1	0	1	1	1

Word embeddings

- Words are represented by numerical vectors called “embeddings” that capture meaning
  - Word2Vec, GloVe
- Pre-trained models use Wikipedia, Google News, Twitter, etc.



Transformers

- **Transformers** have become dominant for text sequence modeling e.g. translation, generation
- Transformers convert text to features via:
  - Word embeddings
  - Positional encodings
  - "**Attention**" – a measure of how strongly words in sentences are related

"The boy didn't study for the test because he was too tired.

outrageously AMBITIOUS

Wrap-up

Duke

PRATT SCHOOL *of* ENGINEERING

Strengths of neural networks

- Ability to model very complicated relationships with a very large number of features. Particularly excel on unstructured data
- Require little to no feature engineering
- Modern tools make it very easy to use neural networks

Issues with neural networks

- Neural networks can be very computationally expensive
- As a result, NNs can be very power-hungry
- Neural networks can be difficult to train
- Difficult to interpret the outputs of deep neural networks
- Easy to overfit, particularly on small data

Wrap Up

- Deep learning adoption has been driven by increasing data and processing power
- Neural networks excel at modeling with unstructured data e.g. images & text
- Should be used with care due to difficulty of interpretability

outraguously
AMBITIOUS

Course Wrap-up

Duke
PRATT SCHOOL *of* ENGINEERING

What we’ve covered

- What machine learning is and how it works
- Steps in the modeling process
- How to evaluate models
- Types of ML algorithms and use cases
- Deep learning & its applications

AI Product Management Specialization

Course 1

Machine Learning Foundations for Product Managers

Course 2

Managing Machine Learning Projects

Course 3

Human Factors in AI