Deep Learning 101

Patrick Donnelly

September 26th, 2019

Who We Are

“We answer the call” has always been our mission - a relentless dedication to responsive service. We answer the call for a full-service IT solution provider delivering best-in-class assessment, design, testing, procurement, integration and support services – customized to meet your needs and requirements.

We will optimize your IT assets, evaluate and implement new technologies and provide a roadmap toward cloud-centric infrastructure models to lower costs, increase agility and increase competitiveness.

Deep Learning in 12 Slides

It's just millions of multiplications, additions and activations

Deep learning (DL) uses artificial neural networks (ANNs) to learn nonlinear functions with many parameters

ANN **inputs** and **outputs** are sets of **nodes** (neurons)

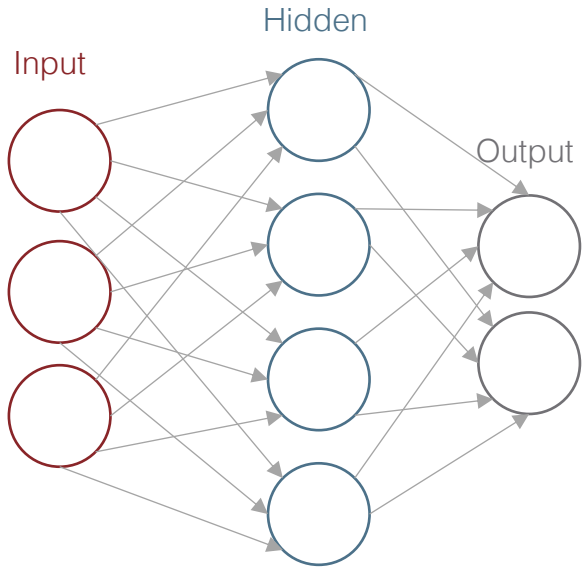
The set of input nodes is the **input layer**

The set of output nodes is the **output layer**

Hidden nodes construct additional layers between inputs and outputs, known as **hidden layers**

A deep neural network (DNN) has more than one hidden layer

Is this a deep neural network?



GROUPWARE TECHNOLOGY

DL Architecture, Weights and Biases

Terminology

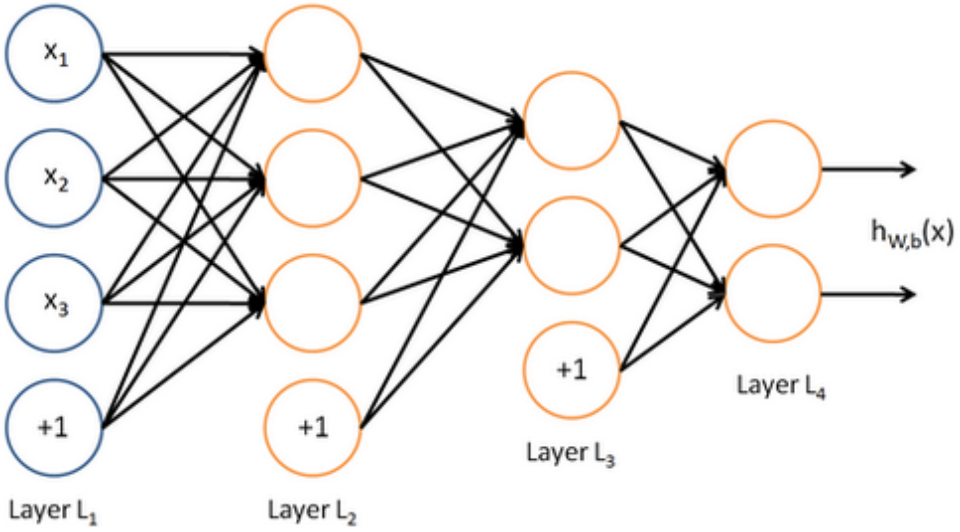
A DL model contains an **architecture** defining the number of nodes and how they're connected.

The output of one layer is the input to the next layer

To get the output of a layer, we multiply the inputs by **weights** and (optionally) add a **bias**

Then we sum the weights and biases

But wait! There's one more thing …



Is this a deep neural network?

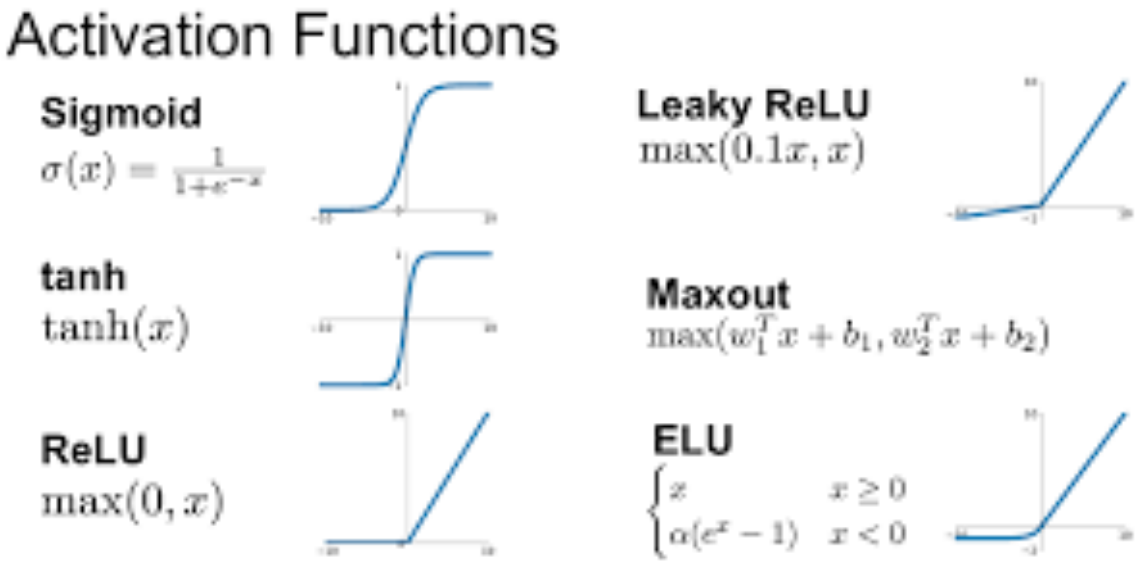
Types of Activation Functions

What do we want from an activation function?

Must be nonlinear (identity no good)

Step functions (only two outputs) problematic for e.g. multiclass classification

Issue with **vanishing gradients** (sigmoid, tanh not ideal in many cases)



ReLU Activation Function

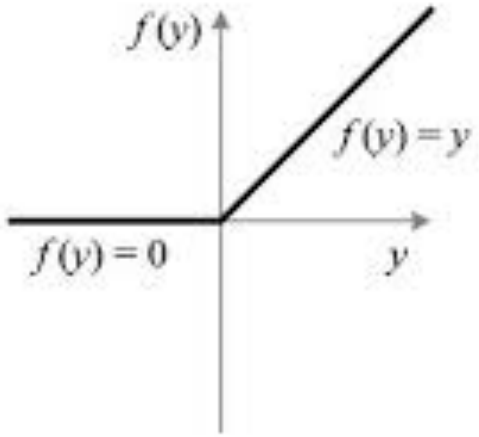
Rectified linear unit and variations

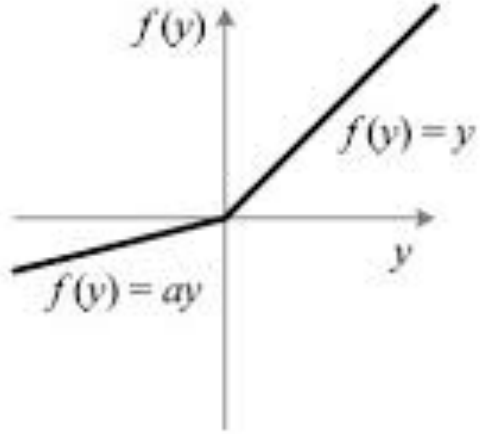
Sets negative values to zero, keeps rest the same

Commonly used for convolutional neural networks (we'll get to these shortly)

Kills gradients for negative values, but not positive values

Leaky ReLU (figure on right) addresses vanishing gradient problem ($0 < a < 1$)





DL Training vs. Inference

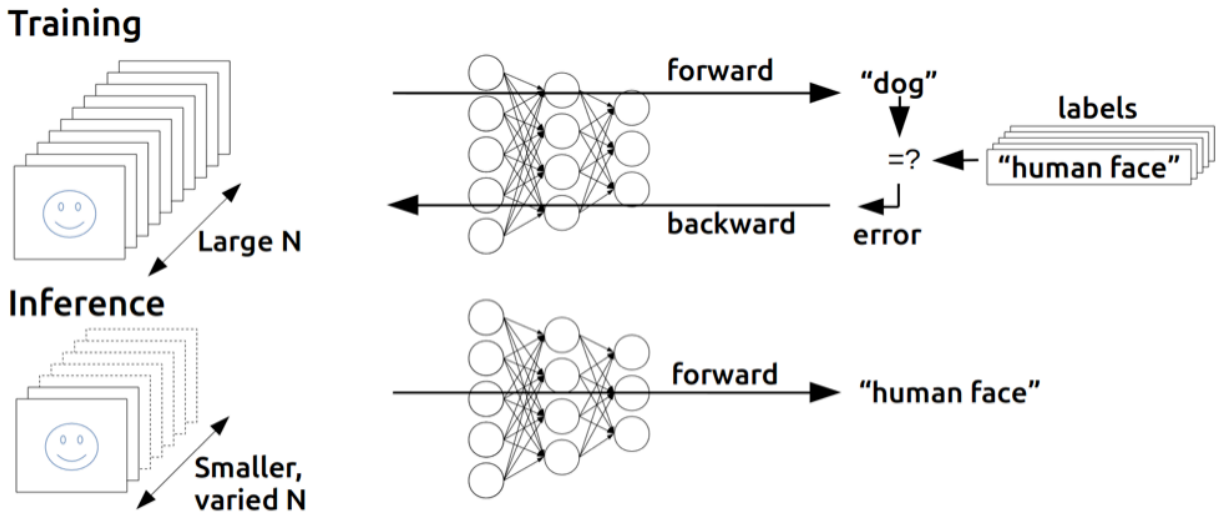
Terminology

Training: learn model from data

Inference: use trained model to make predictions on unlabeled data

Training needs lots of (often labeled) data

Inference can be done on as little as one observation and often requires low latency



Types of Deep Learning

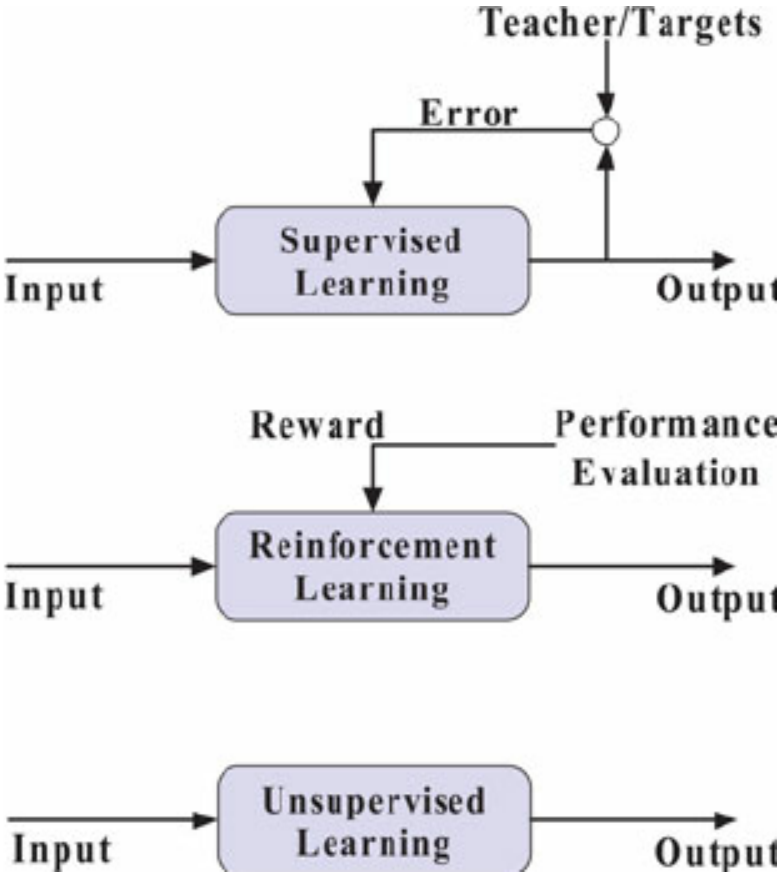
Terminology

Supervised learning: feed labeled data to model, compare predicted labels to actual labels, update model, repeat

Unsupervised learning: feed unlabeled data to model, extract features, update model, repeat

Reinforcement learning: agent acts in environment, receives reward, updates state

There's also semi-supervised learning, which learns features from labeled and unlabeled data



Convolutional Neural Networks (CNNs)

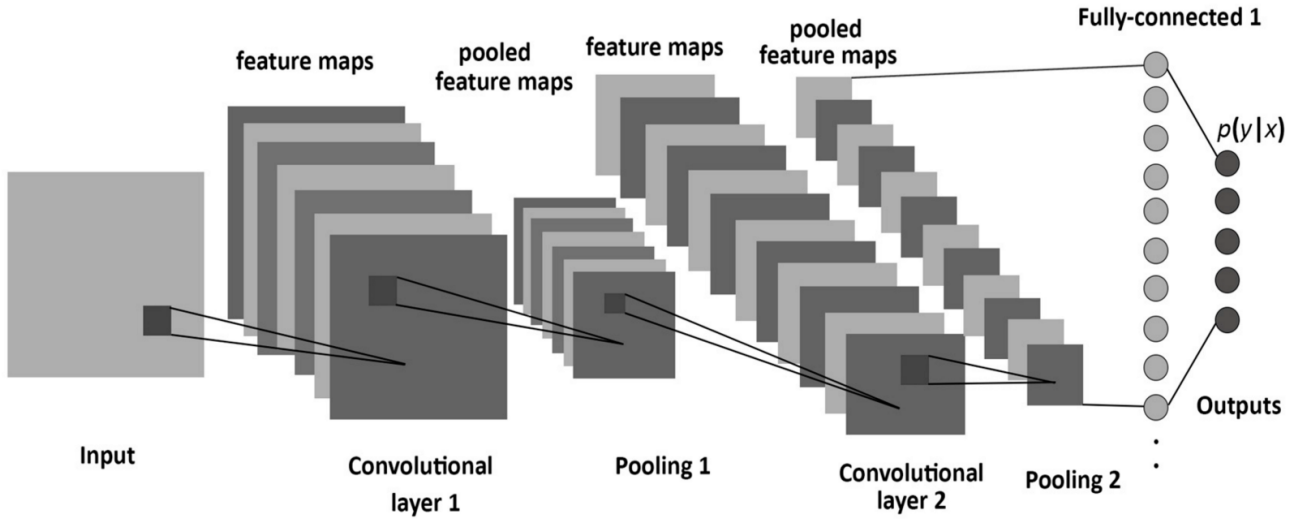
How to teach a computer to see

In a **fully-connected neural network**, all nodes in one layer are connected to all nodes in the next layer

In contrast, **CNNs** learn a set of filters or **kernels** of shared weights

The dot product of kernels and inputs generates feature maps

These feature maps are concatenated into fully-connected layers and downsampled



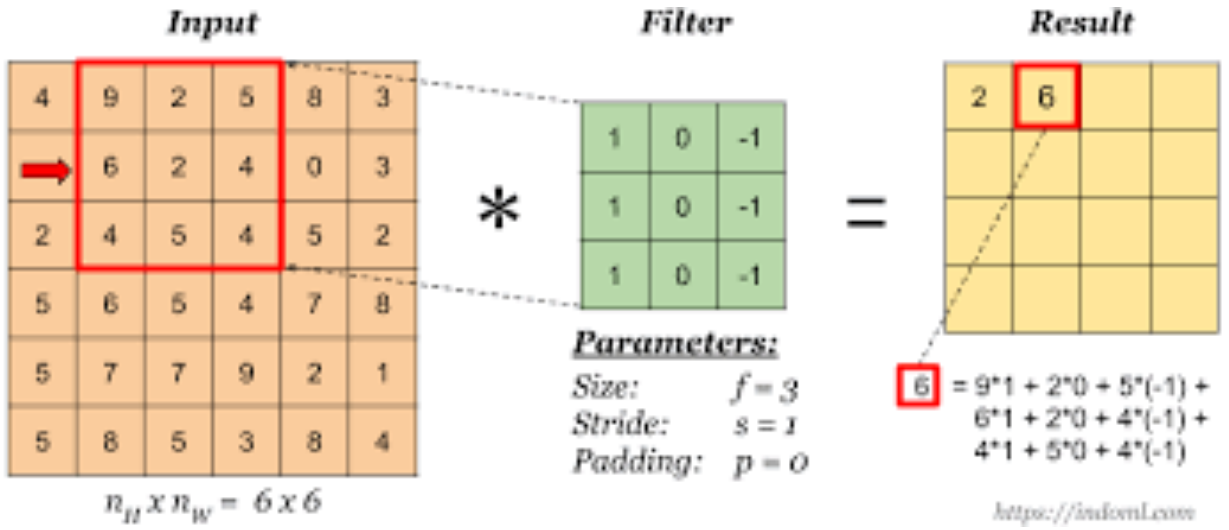
What is a Convolution?

It's just a dot product

To get the output of a convolution:

Starting in the upper left of the input, multiply
each value of the kernel by the value of the corresponding entry in the input

Stride the kernel across the input, first left to right, then downward, repeating the operation at each position



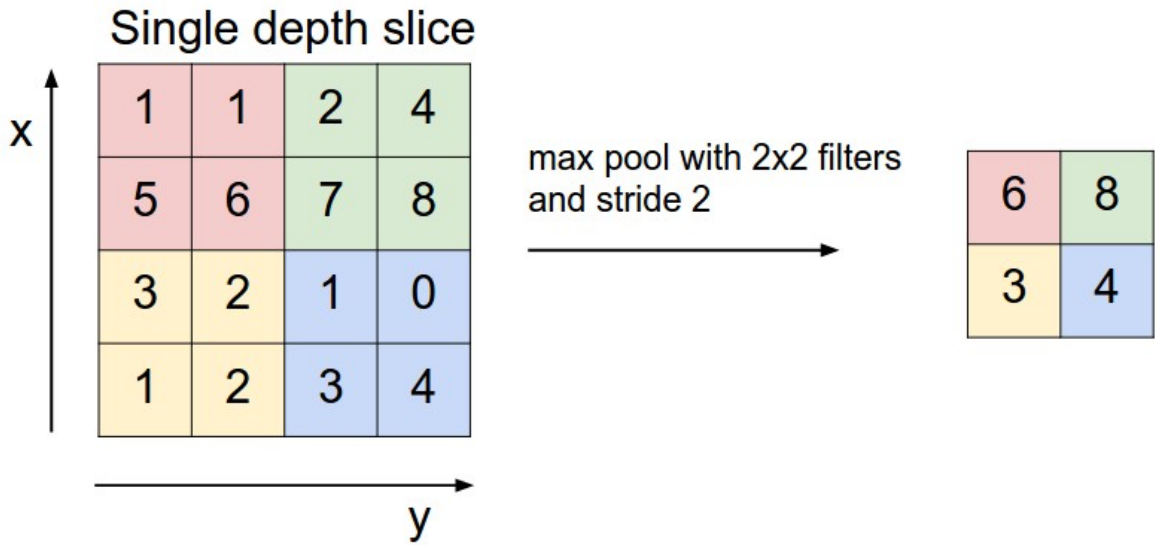
We're Not Done Yet!

Pooling and other stuff

Convolutional operations are commonly followed by (typically) max or average **pooling** over a specified field

Max pooling takes a (typically 2D) region as an input and outputs the maximum value

Other tricks include **local response normalization** (AlexNet),**inception modules** (GoogleNet) and **residual connections** (ResNet)



So How Do These Networks Learn?

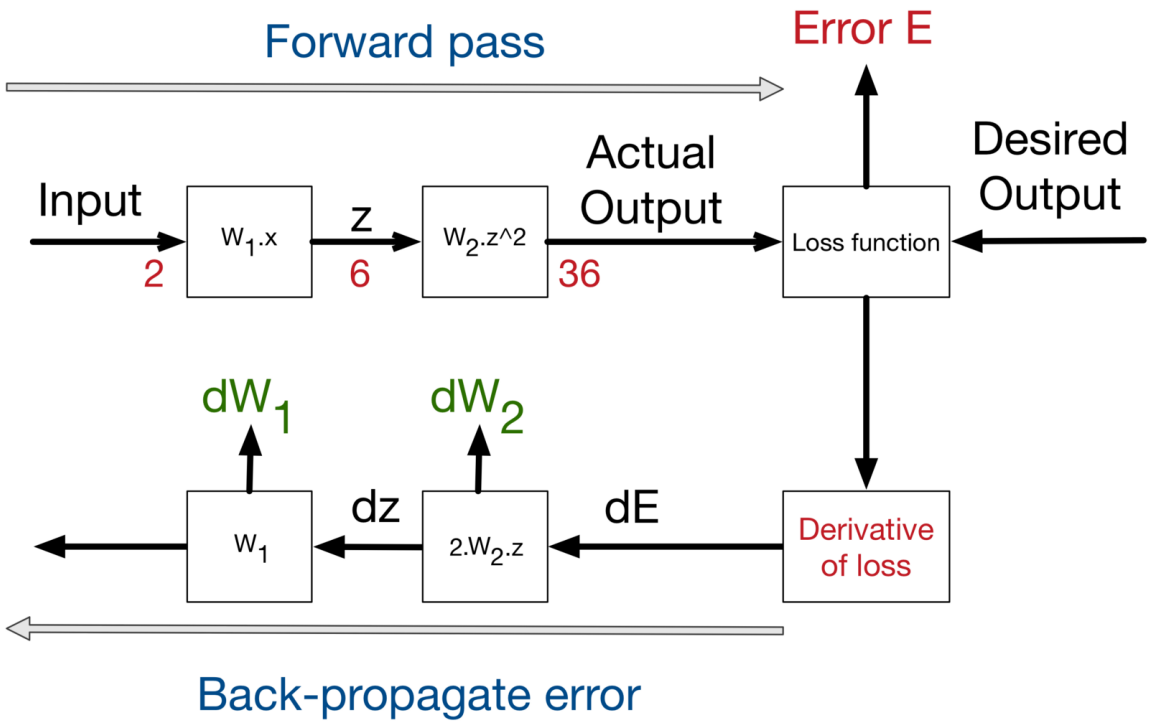
Forward and backward propagation

Step 1: • Compute output of network (**forward propagation**)

Step 2: • Compare output of network with label

Step 3: • Update weights and biases (**backpropagation**)

Step 4: • Repeat until desired accuracy is attained!




Questions?

THANK YOU

GROUPWARE TECHNOLOGY

We Answer the Call