Deep Learning Crash Course

Maxime Rousseau

Resident - Division of Dentistry, MCH

January 7th, 2021

Disclaimer: I am not a computer scientist. Some of the information in this presentation may be incomplete/incorrect.

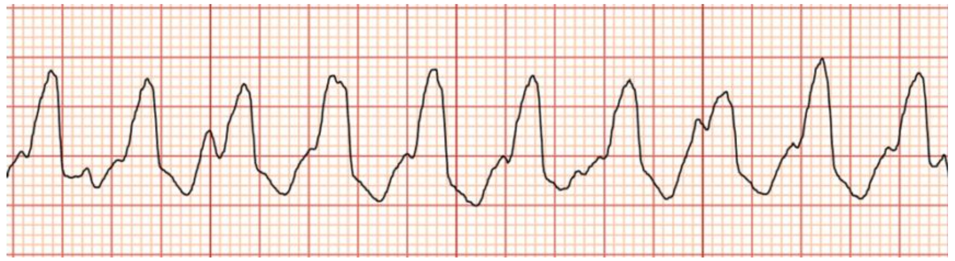
From Statistics to Deep Learning

- Statistics to draw inference from data (linear regression, hypothesis testing, etc.)
- Machine learning is used make predictions (**classification**, recognition, language translation, generations, etc.)
- Deep learning as a subfield of machine learning to generate high level abstraction from large dataset

Types of Learning

1. **Supervised**: we already know the answer to the problem (i.e., labeled images)
2. Unsupervised: the model must learn to organize data without labels

Input (x)



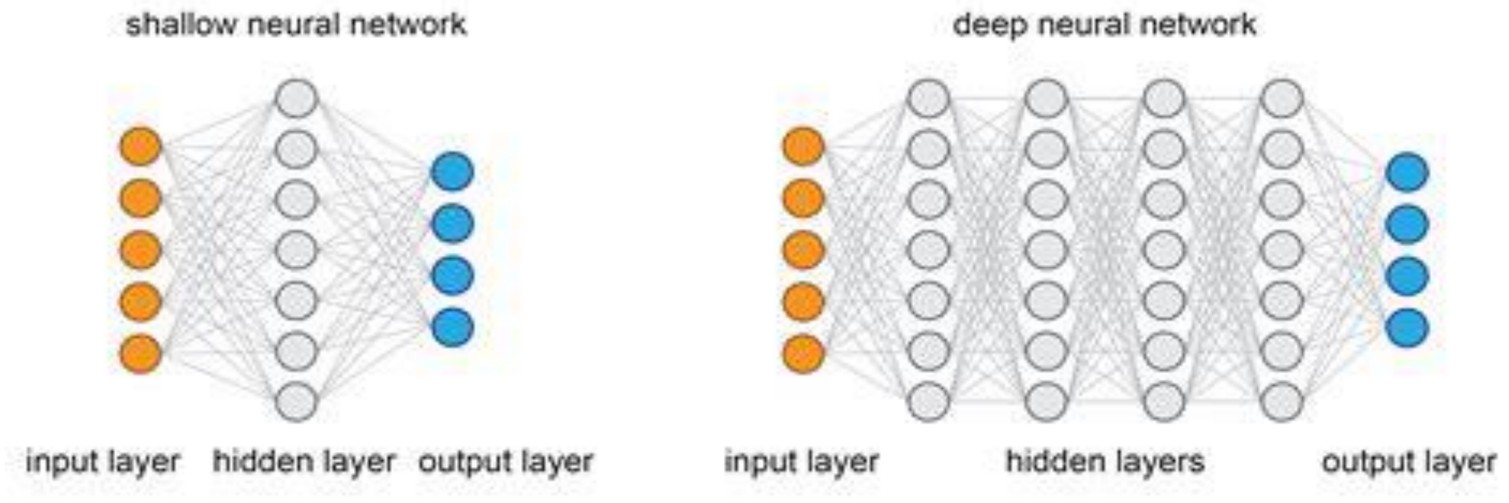
Label (y)

Ventricular tachycardia

What is Deep Learning

– neural network with multiple **hidden layers**

– understanding **features** (or high-level abstractions) from the dataset



How does a Deep Neural Network *Learn*?

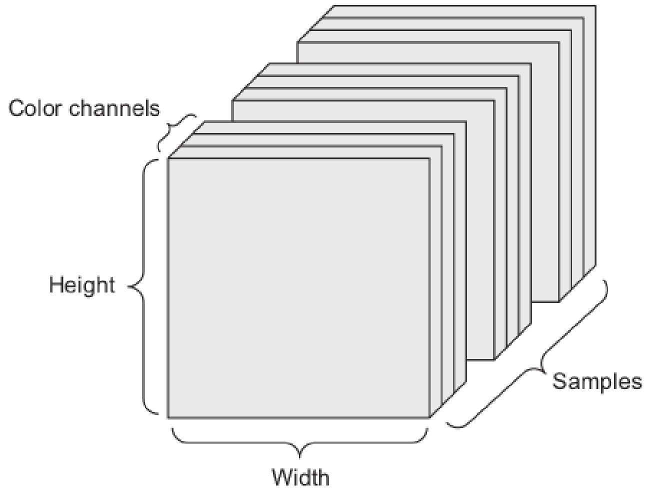
A few components are required:

- Tensors and tensor operations (dataset – inputs and labels)
- Neurons (the model)
- Backpropagation (the mechanism)

Tensors and Tensor Operations

Tensor are simply arrays of digits they can have a variety of forms/dimensions depending on the type of data.

- 1D tensor: list of metrics from a patient
- 2D tensor: facial image from a patient
- 3D tensor: video data or 3D model



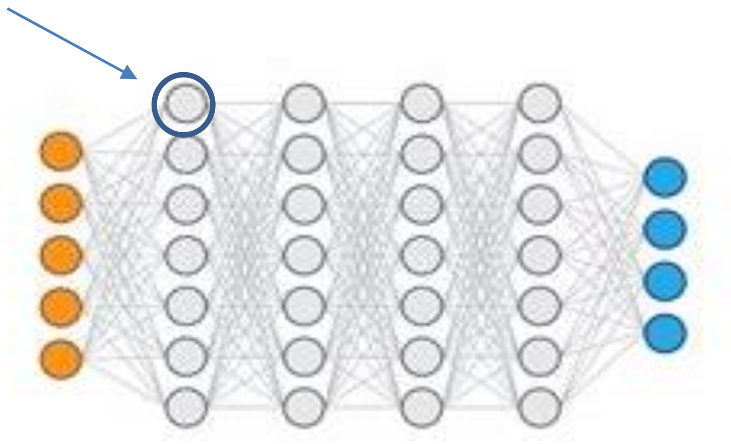
We can perform tensor operations such as addition, multiplication, dot product to transform our input data.

The Neuron

Deep neural networks = chaining tensor operations to transform input data

A basic neuron

$$Y = ReLU(w \cdot X + b)$$

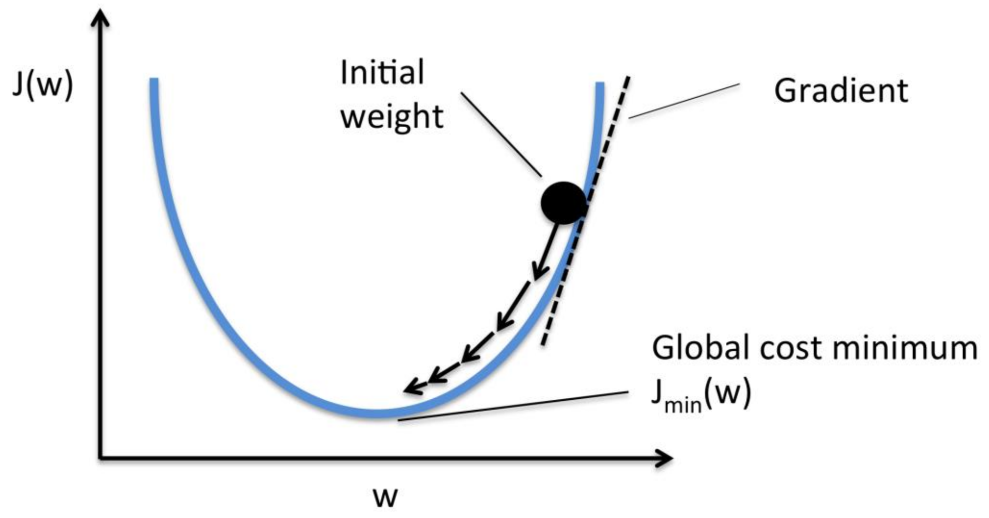


- **b:** bias
- **w:** weights
- **X:** input (tensor)
- **Y:** output (label)
- **ReLU:** an activation function (Rectified Linear Unit)

Backpropagation

A feedback mechanism that allows the network to learn during training.

- loss function (compute how well the network performed on a subset of the training data)
- gradient descent (adjust the network parameter based on the output of the loss function)

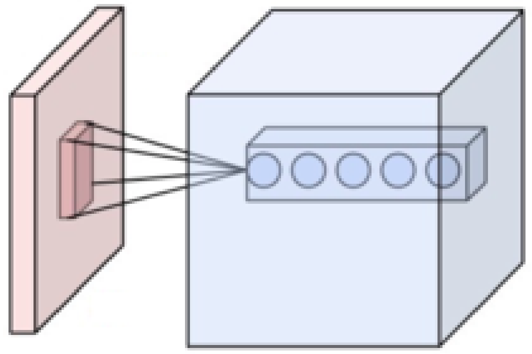


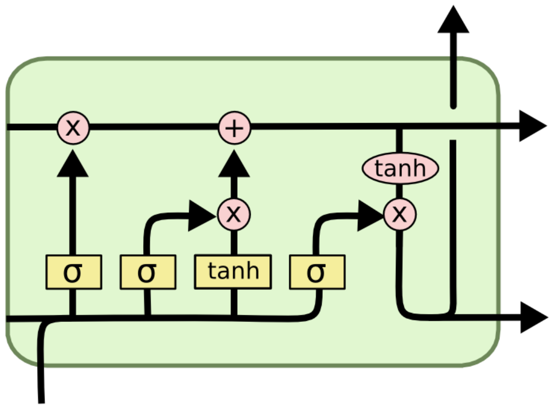
This is done by chaining the derivatives of the tensor operations to compute. There are multiple functions that exist to compute loss and perform the weight update (gradient descent).

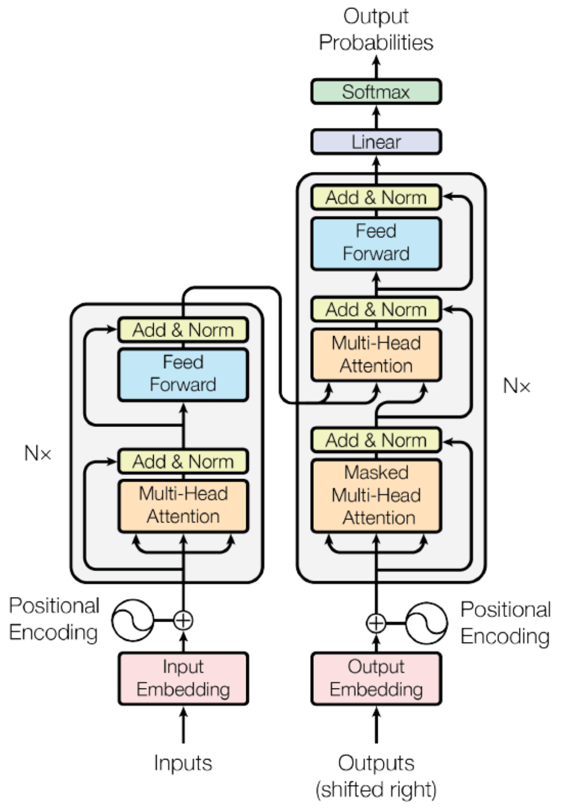
Architectures

Beyond backpropagation and fully connected layers:

- Convolutions (images)
- Recurrent networks: LSTM, GRU (text)
- Transformers and attention mechanisms







In Practice

1. Material/Setup
2. Dataset
3. Workflow
4. Potential issues

Material/Setup

Requires basic understanding of programming to start deep learning: basic syntax, control flow, *data structures*, etc.

- Python 3.7+
- Jupyter notebook
- Tensorflow/*Keras*
- GPU vs cloud compute

Deep learning is code, you need to be able to debug your scripts and prepare your data.

Defining a Goal

Consider before gathering data:

- what real problem are we trying to solve?
- what type of deep learning task does this problem fall into?
- what information is required to solve the problem?

Datasets

*. <u>Data Preprocessing:</u> Computers can only understand numbers. Preprocessing is the transformation complex data types into tensors which can be used by a model.

1. <u>Distribution:</u> how to create a good dataset → balanced, out of sample distribution
2. <u>Sample size:</u> what is a minimum sample size? will depend on how complex the data is (i.e., higher dimension will tend to require more data)
3. <u>Split:</u> training, validation and test (and how each are created and used)

Workflow

Projects tend to be iterative in deep learning. A lot of experimentation is usually required in order to create a model with good performance.

1. Data preprocessing (view and get familiar with your dataset)
2. Define model (start with the simplest possible model, which will be easier to debug)
3. Validate and tune hyperparameters
4. Test model and evaluate performance
5. Revise model

Potential Issues

– Overfitting → data augmentation, architecture changes, etc.
– Debugging →check dimensions
– Lack of data → Transfer learning

Tip: Set a clear goal and determine the data type you can use. Build a small dataset and begin experimenting. Begin with the simplest form of the problem you wish to solve and as you gain experience and become more comfortable with deep learning.

Andrej Karpathy's – Recipe for Training Neural Networks: https://karpathy.github.io/2019/04/25/recipe/