ML Tutorial Class

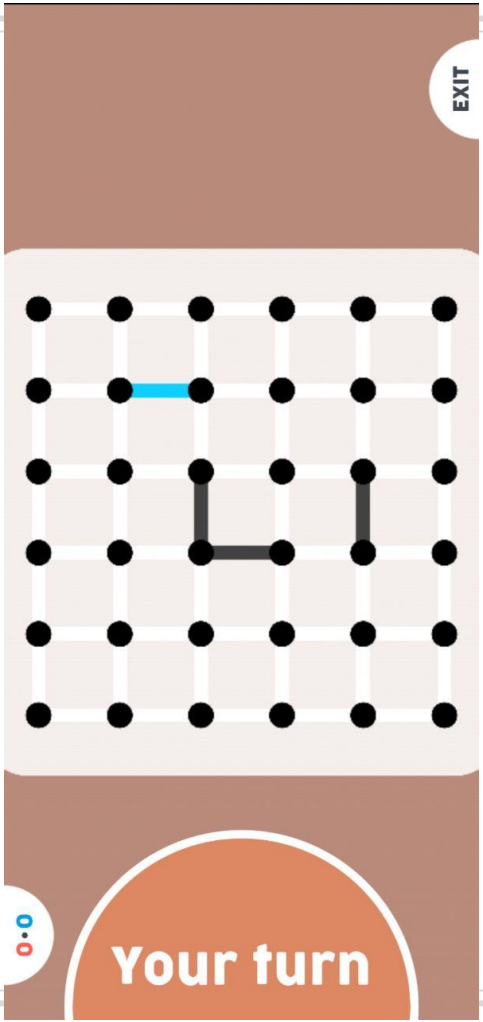
IMT2019525 VIJAY JAISANKAR

Agenda

In today's session, we will cover...

- Neural networks
  - Why
  - How
  - Where
  - When
  - Who
  - What
  - あれは何ですか。

Have you ever played this game?


EXIT
0 • 0
Your turn

Does this have any resemblance with Graphs?

Game: example

- Are there *nodes*?
- Are there *connections* between nodes?
- What if we have multiple *layers* of boards?

Now, let’s look at the brain.

Or, maybe not :)

Have you seen this meme?

Do you see how neurons are connected to each other?

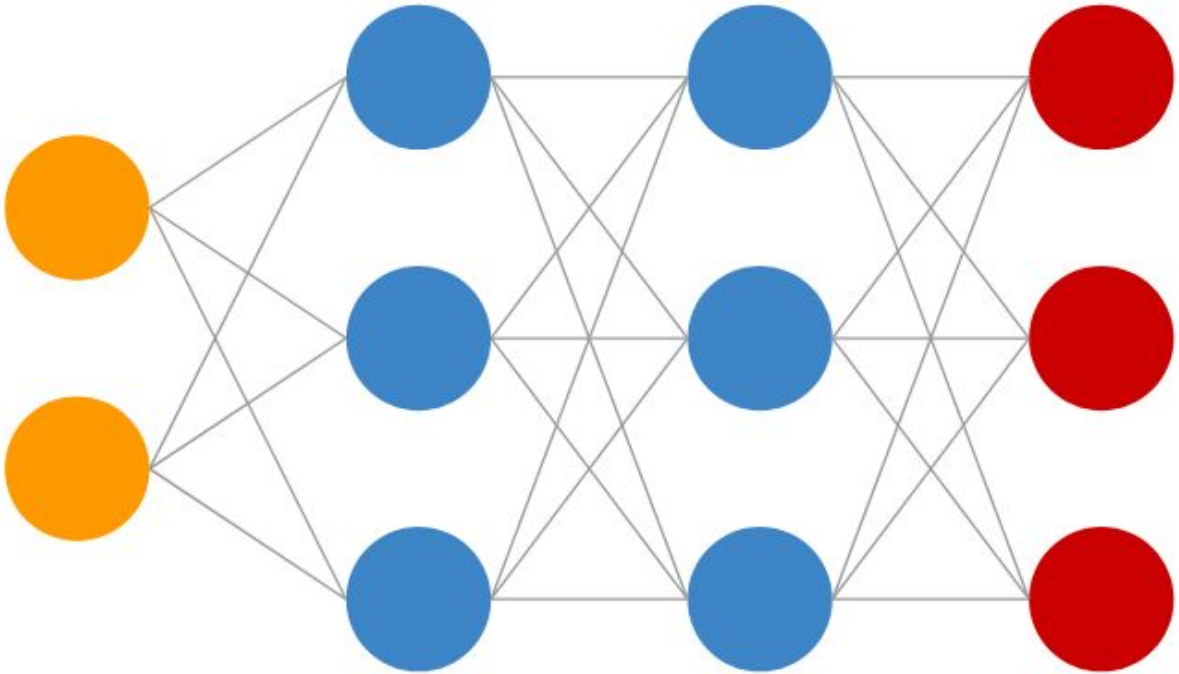
Neural networks are modelled after human brains.

*But, what **is** a neural network?*

Introduction to Neural networks

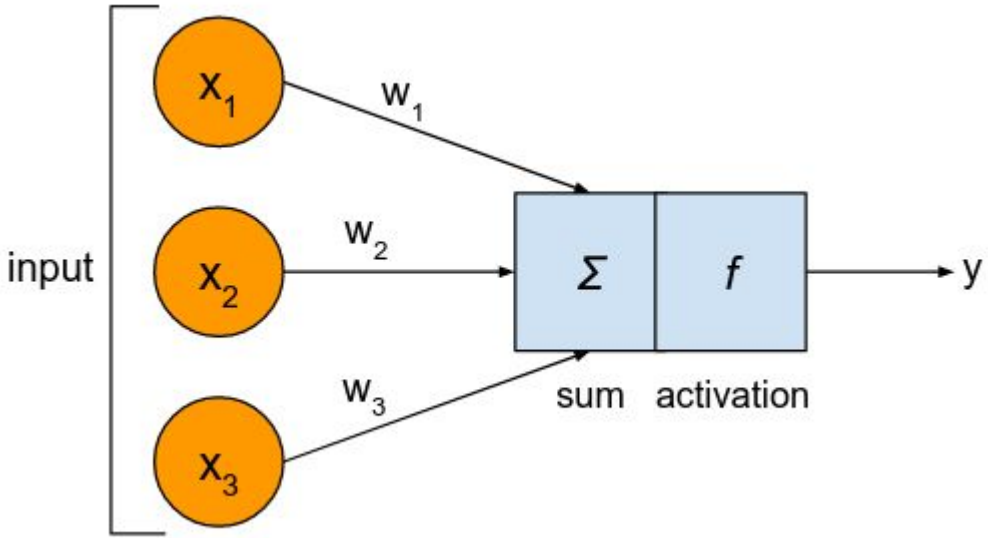
- Nodes arranged in layers
  - Neurons: computational units
- Connections between different layers

A simple neural network



We have made a neural network before!

Logistic Regression as a small NN



If *activation* is the sigmoid function, what does this look like?

- Binary classification - logreg!

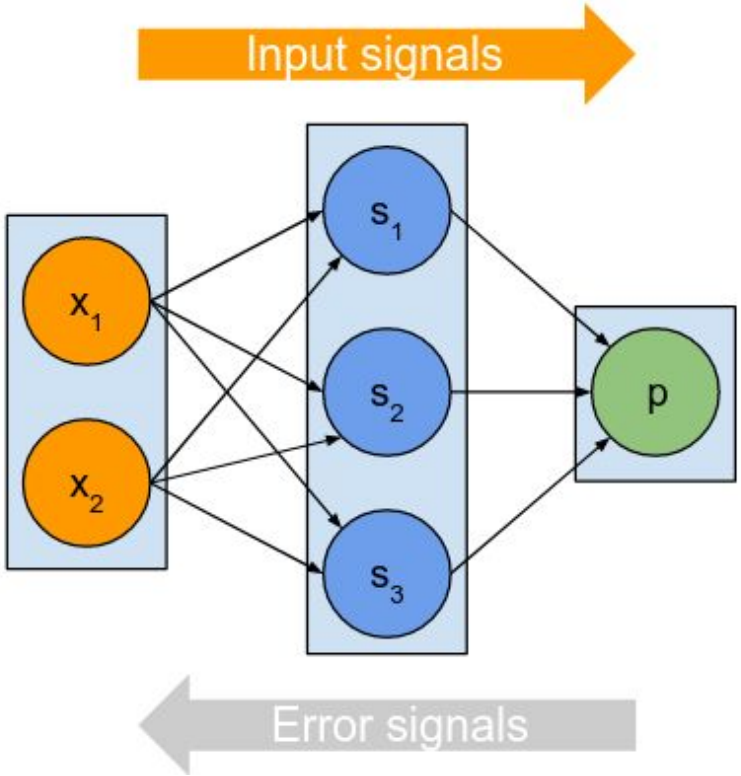
But, wait...

Question

- Are we missing something?
- M_1 (M_2 (M_3 … (X) …)) is still linear!
- We need to add non-linearity to the model…
  activation functions

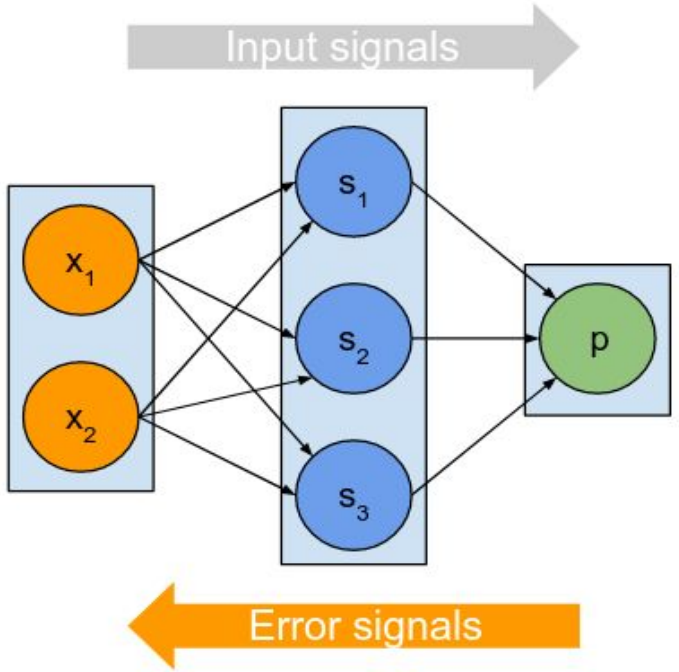
Let’s get to the fun stuff… how would you train an NN?

Credits: Valerio Velardo



1. Get prediction
2. Calculate error
3. Calculate gradient of error function over the weights
4. Update parameters

Forward pass



1. Get prediction
2. Calculate error
3. Calculate gradient of error function over the weights
4. Update parameters

Backward pass

Backpropagation

https://github.com/musikalkemist/DeepLearningForAudioWithPython/blob/master/7-%20Bagkpropagation%20and%20gradient%20descent/slides/Training%20a%20neural%20network_%20Backward%20propagation%20and%20gradient%20descent.pdf

Let’s play around with neural networks!

https://playground.tensorflow.org

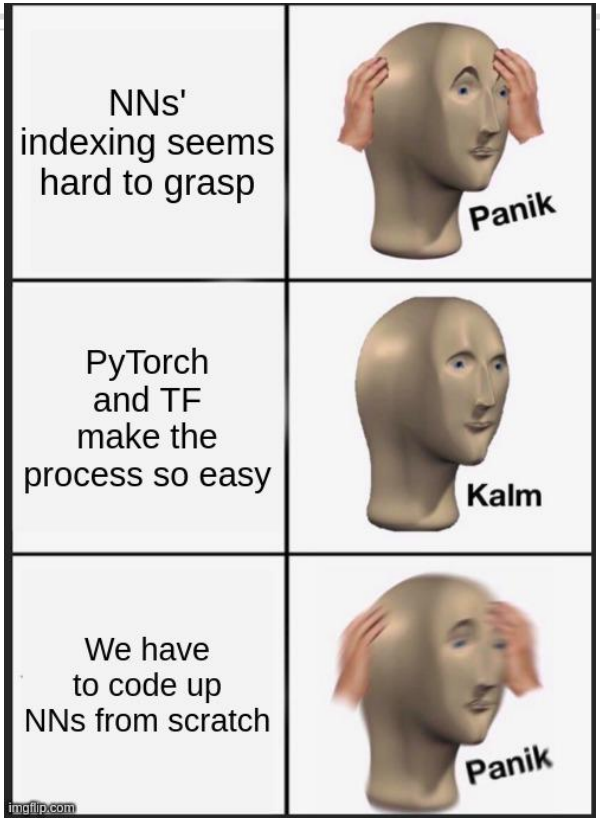
Some things to consider while training Neural networks

Miscellaneous points

- Handling underfitting
  - Deep learning typically needs a lot of data
- Overfitting can be notorious!
  - Dropout layers
- Deciding on a model architecture
  - Very hard
- Building on existing models
  - Transfer Learning

Let’s build a neural network in PyTorch!

https://pytorch.org/tutorials/beginner/basics/buildmodel_tutorial.html



If you can code up NNs, a lot of the matrix manipulations will become very familiar to you, and this will be helpful going forward.

Thank you!