Fast Neural Network... a no brainer!

RICCARDO TERRELL

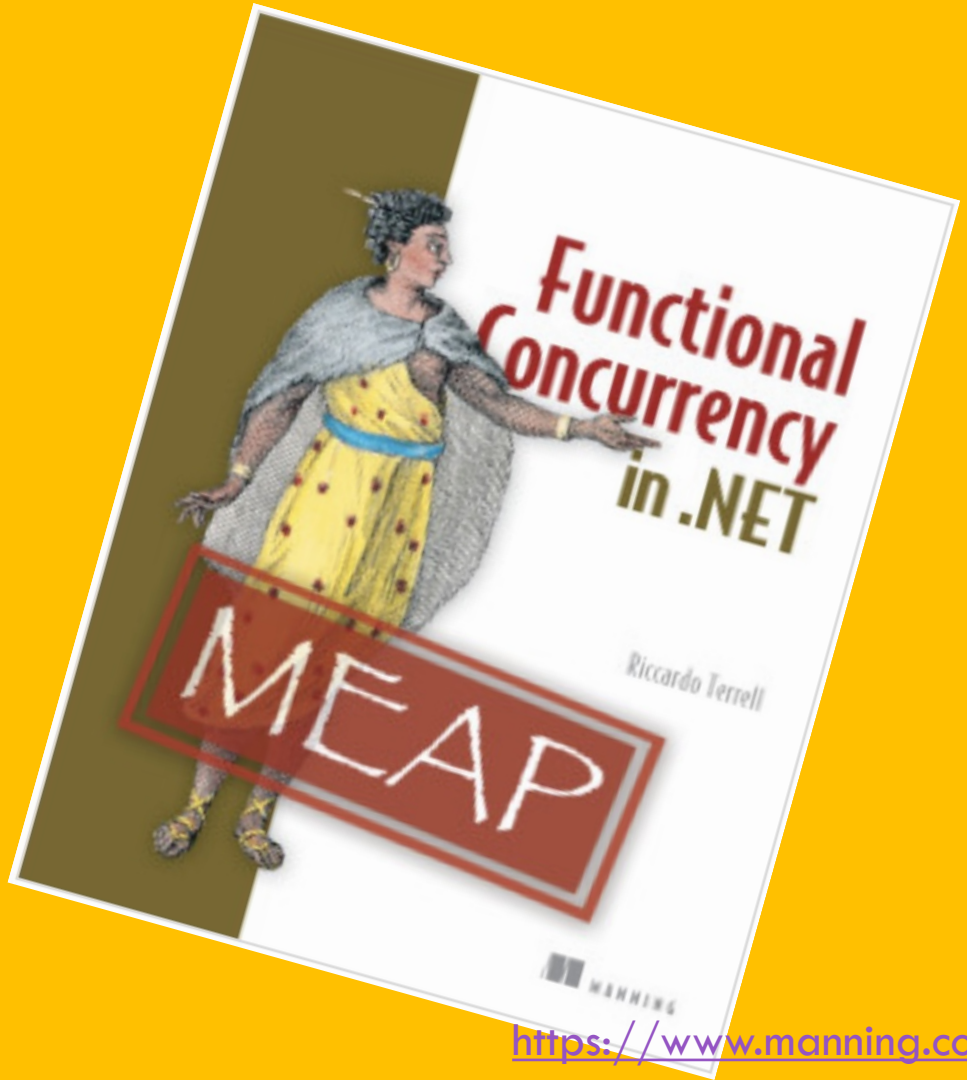
Agenda

- What types of problems does a neural network solve?
- What exactly is a neural network?
- How does a neural network actually work?
- Demonstrate How to Parallelize a Neural Network
- Demonstrate a Neural Network in Action

Introduction - Riccardo Terrell

- Originally from Italy, currently - Living/working in Washington DC ~10 years
- +/- 19 years in professional programming
  - C++/VB → Java → .Net C# → Scala → Haskell → C# & F# → ??
- DC F# User Group - Organizer
- Working @ statmuse
- *Polyglot programmer – I believe in the art of finding the right tool for the job*

@trikace *rickyterrell.com* *tericcardo@gmail.com*



Use code coupon

terrellug

for 39% off

Expiration 20 February

https://www.manning.com/books/functional-concurrency-in-dotnet


What about you?

Goals

Neural Networks ease complex problem solving…

Neural Networks are based on simple linear algebra!

Neural Networks can be parallelized!!

…to be very fast

Machine Learning Algorithms

- *Linear Regression*
- *Logistic Regression*
- *Decision Tree*
- *SVM*
- *Naive Bayes*
- *KNN*
- *K-Means*
- *Random Forest*
- *Dimensionality Reduction Algorithms*
- *Gradient Boost*
- *Adaboost*

Supervised Learning

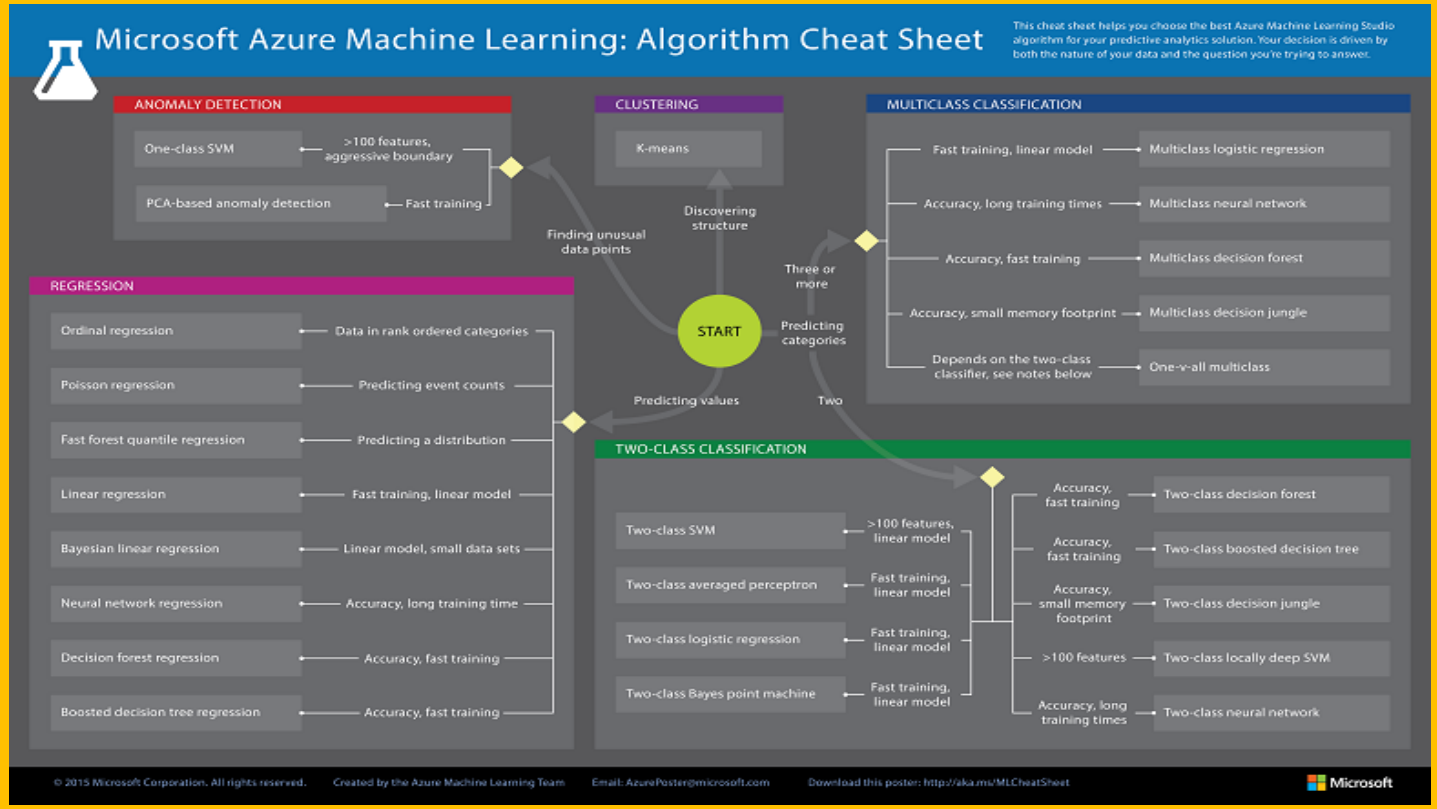
Unsupervised Learning

Unsupervised & Supervised Learning

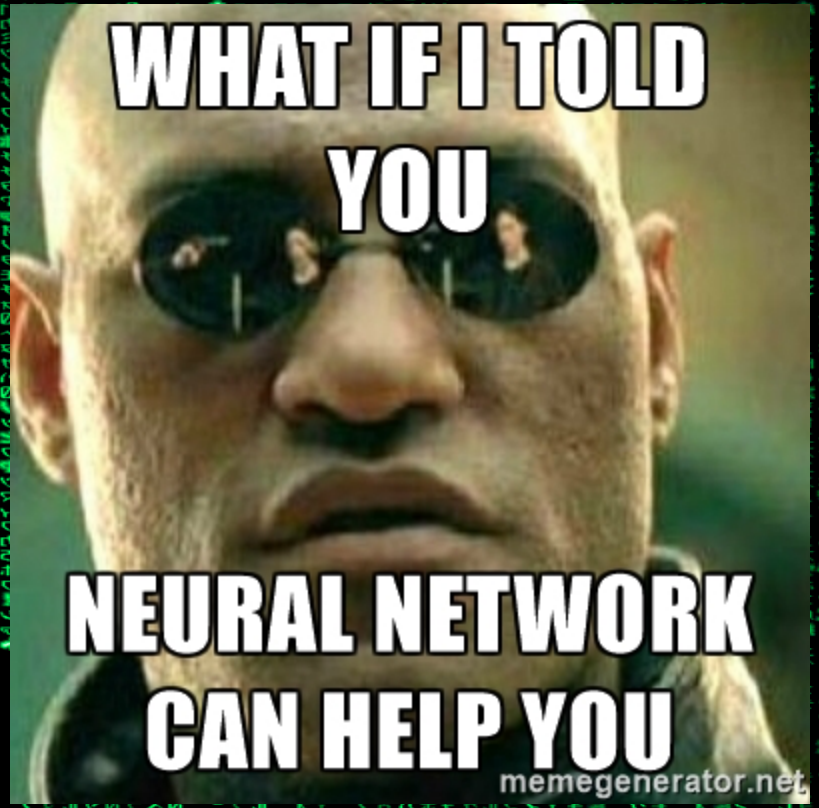
- unsupervised learning:
  - gaussian mixture models
  - latent semantic analysis (better pca for text)
  - self organizing maps
  - smarter interfaces (API to deal with reduction chaining)
  - multi-processor, gpgpu

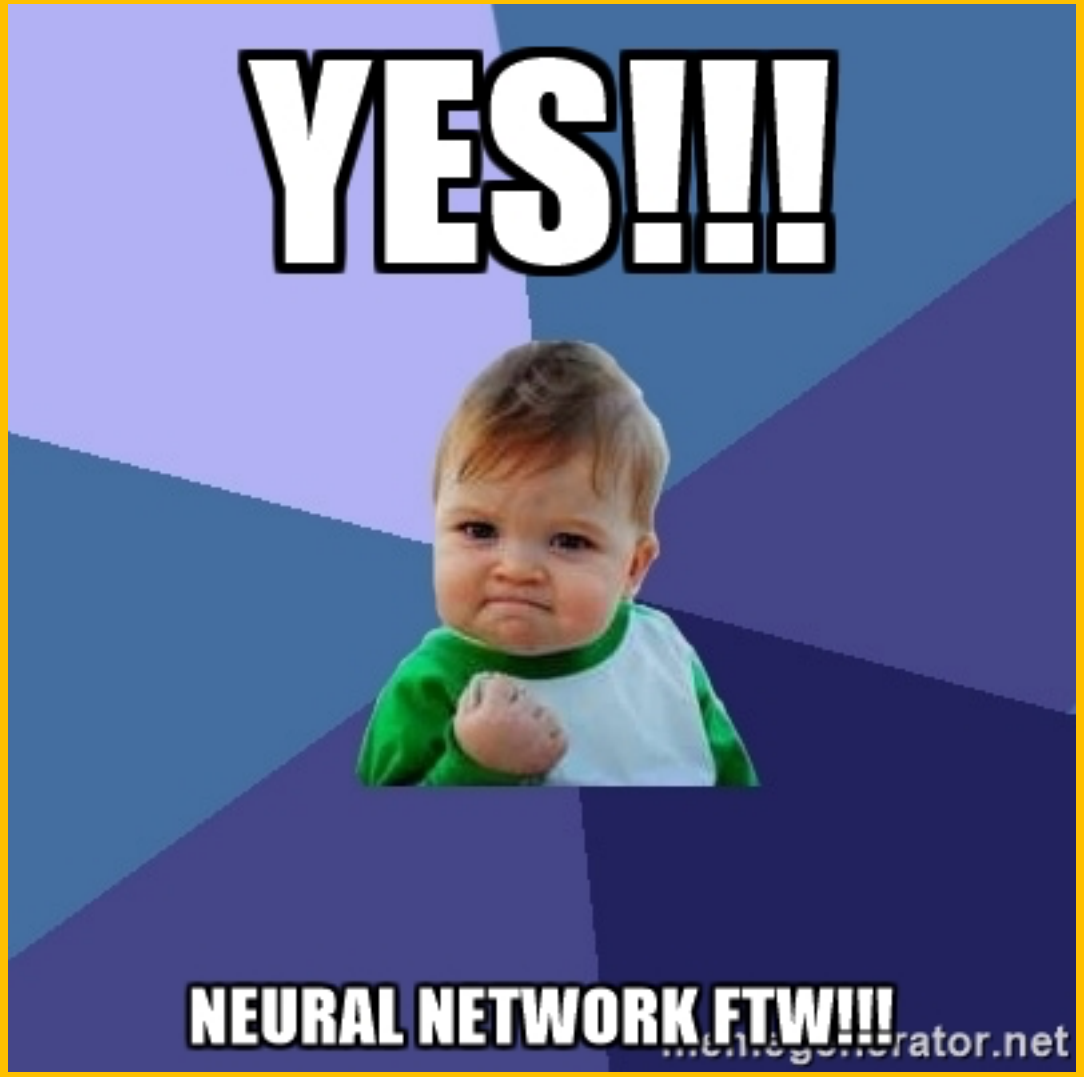
- supervised learning:
  - support vector machines (non-linear classification models)
  - stochastic models (naïve bayes, etc.)
  - neural networks
  - simplified interface (system chooses appropriate model by interrogating label)
  - probabilistic graphical models
  - multi-processor, gpgpu

Microsoft Azure ML Cheat Sheet



https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-cheat-sheet


WHAT IF I TOLD YOU
NEURAL NETWORK CAN HELP YOU
memegenerator.net


YES!!!
NEURAL NETWORK FTW!!!

Neural Networks are *multipurpose*

A Neural network is an alternative to:

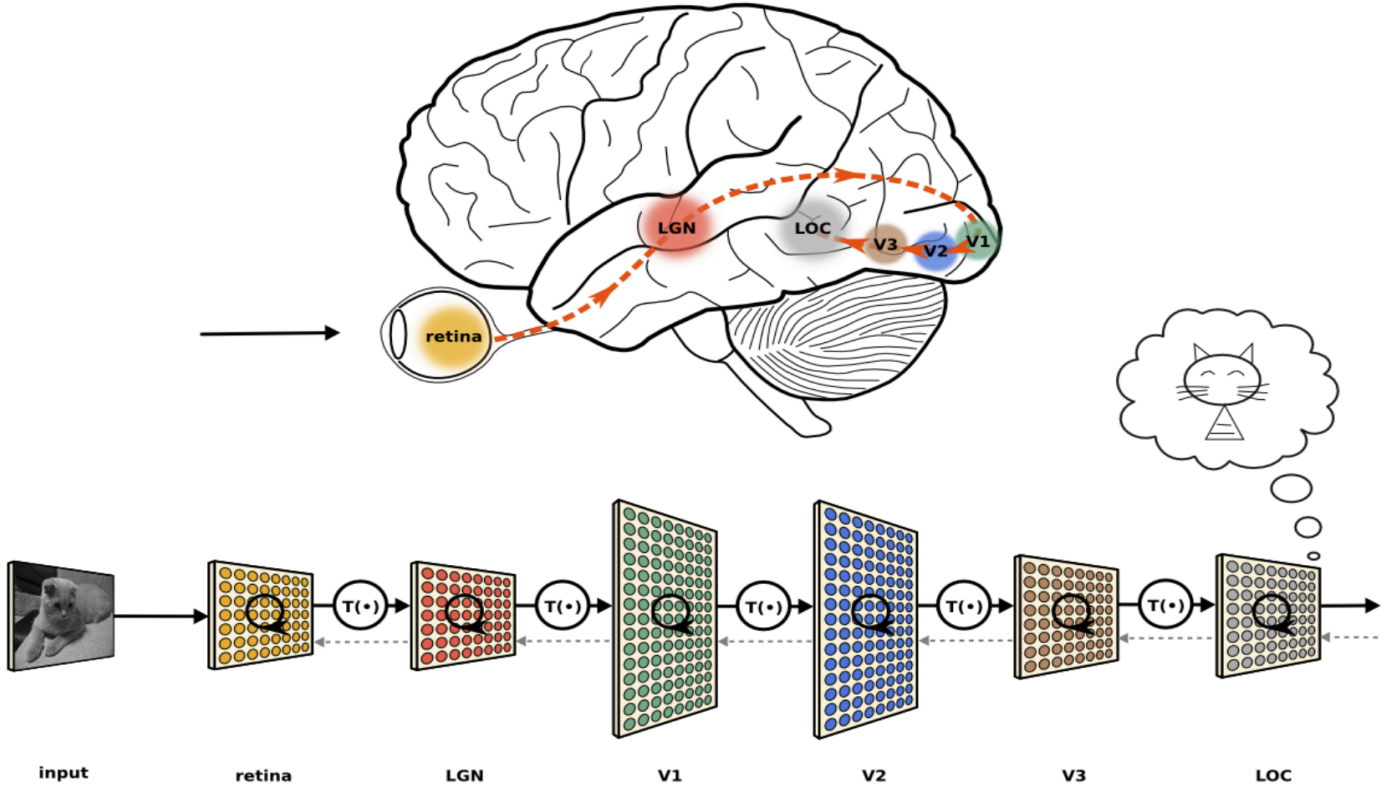
1. *Linear regression:*
2. *Logistic regression*
3. *Naive Bayes*
4. *Decision trees*
5. *Adaptive boosting*
6. *Support vector machines*

What kind of problems does a Neural Network solve?

Problems NN solve

- *OCR (optical character recognition)*
- *Natural language processing*
- *Machine translation*
- *Biology & medicine*
- *Robotics (Autonomous Systems)*
- *Pattern Recognition Time Series Prediction*
- *Signal Processing*
- *Control*
- *Soft Sensors*
- *Anomaly Detection*

What is this?


LGN
LOC
V3
V2
V1
retina
T(•)
T(•)
T(•)
T(•)
T(•)
input
retina
LGN
V1
V2
V3
LOC

NN Prediction Example

Grade	GPA	Age	Gender	Sport
A	3.5	16	M	Baseball
C	2.0	12	M	Football
F	2.1	13	F	Soccer
B	3.5	17	M	Baseball
D	2.0	18	F	Soccer
A	3.8	15	M	Baseball
D	2.3	14	F	Baseball
B	3.3	17	M	????

Training data

features
gpa, age, gender, sport

values (x)
[B, 3.3, 17, M, Baseball]

Independent variables/predictors/ attributes/regressors/x-values

"The thing to classify (predict)"/ dependent variable/y

Recommender systems

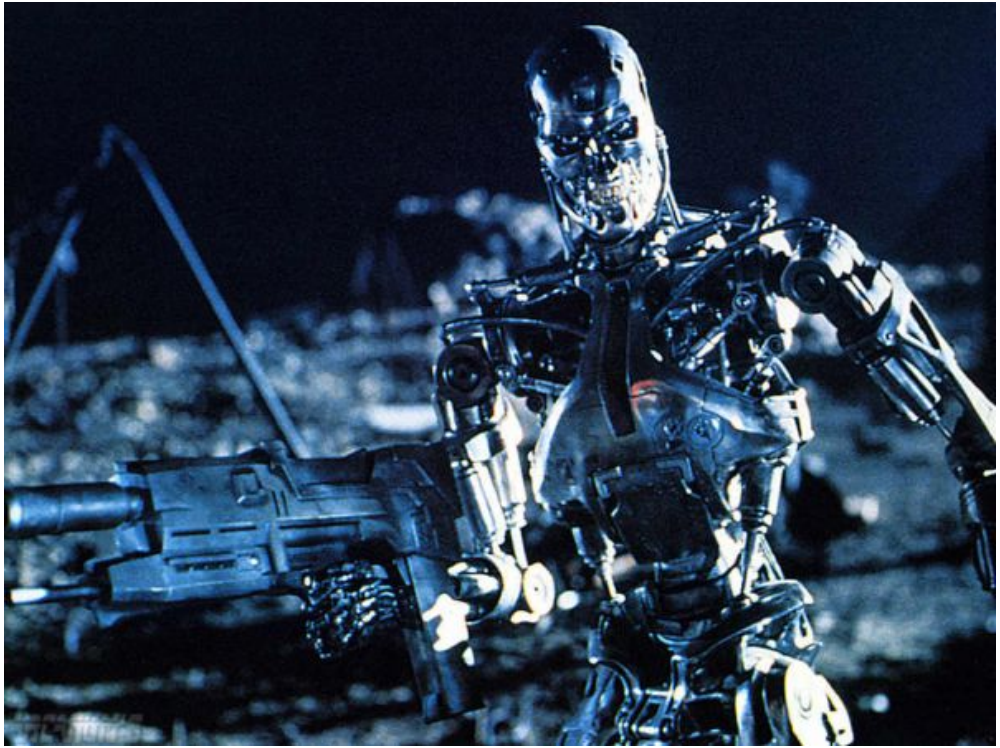
- what do people like?

Recommender systems

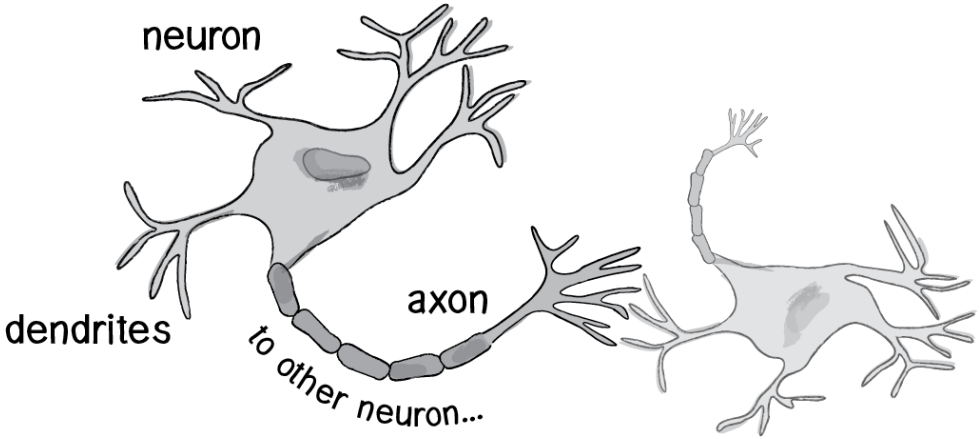
- how are things alike?

What is a Neural Network

What is this Neural Network?



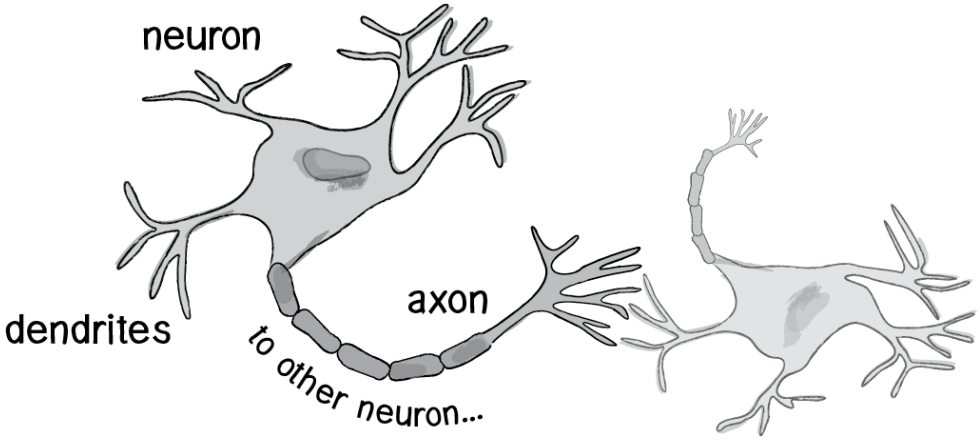
What is Neural Network

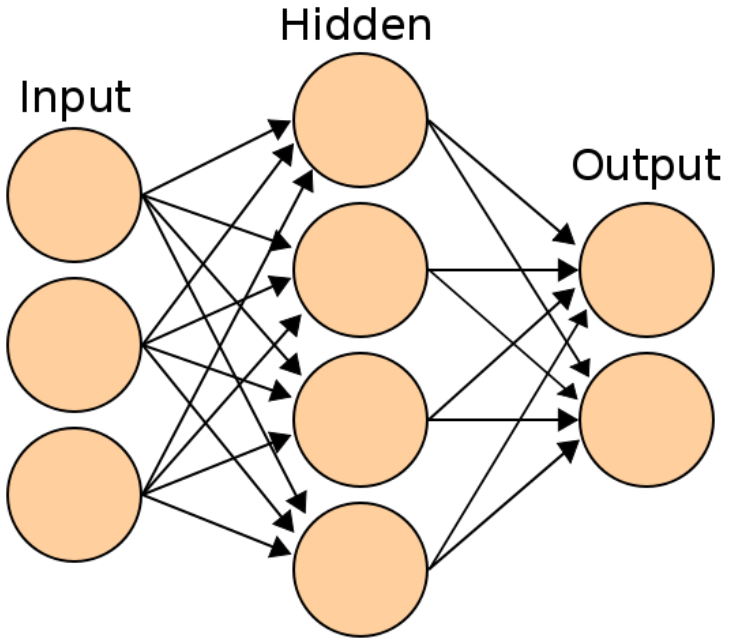


Four parts of a typical nerve cell :

- **DENDRITES:** Accepts the inputs
- **SOMA** : Process the inputs
- **AXON** : Turns the processed inputs into outputs.
- **SYNAPSES** : The electrochemical contact between the neurons.

What is Neural Network





- A neuron: many-inputs / one-output unit
- output can be *excited* or *not excited*
- incoming signals from other neurons determine if the neuron shall excite ("fire")
- Output subject to attenuation in the *synapses,* which are junction parts of the neuron

Mathematical Background

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 2 & 5 \end{bmatrix}$$

Matrix Operations

$$\sum_{i=1}^{n} c = nc$$

Sigma Notation

$$\frac{d}{dx} x^2 = 2x$$

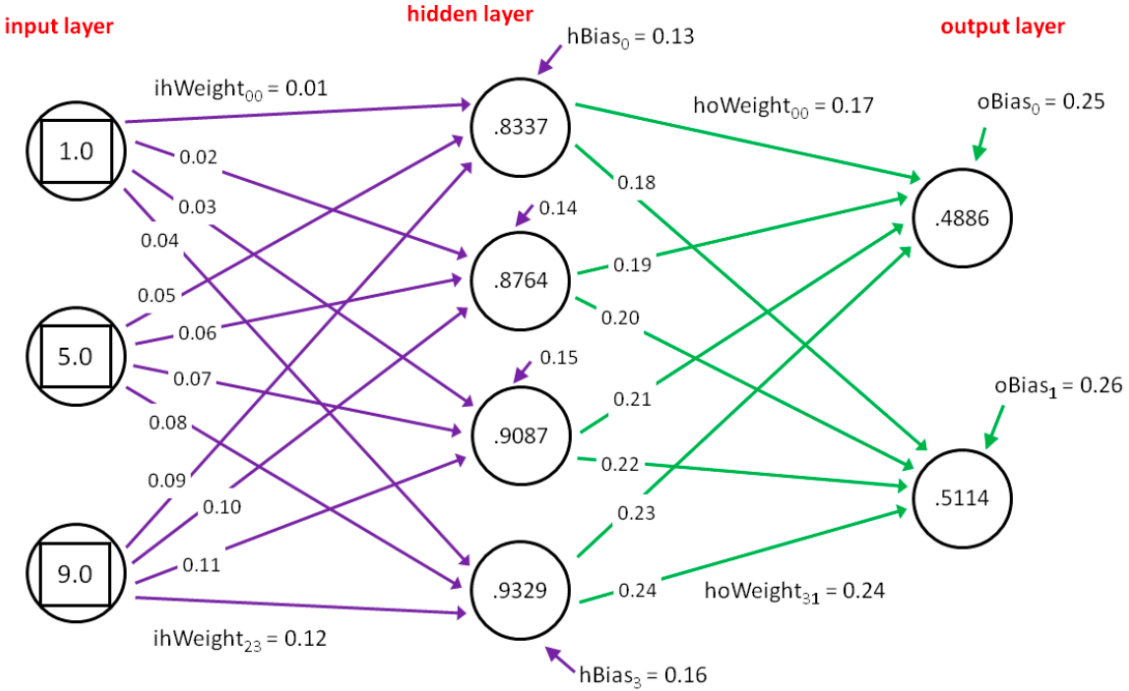
Equation for derivatives and Integrals

Some vocabulary

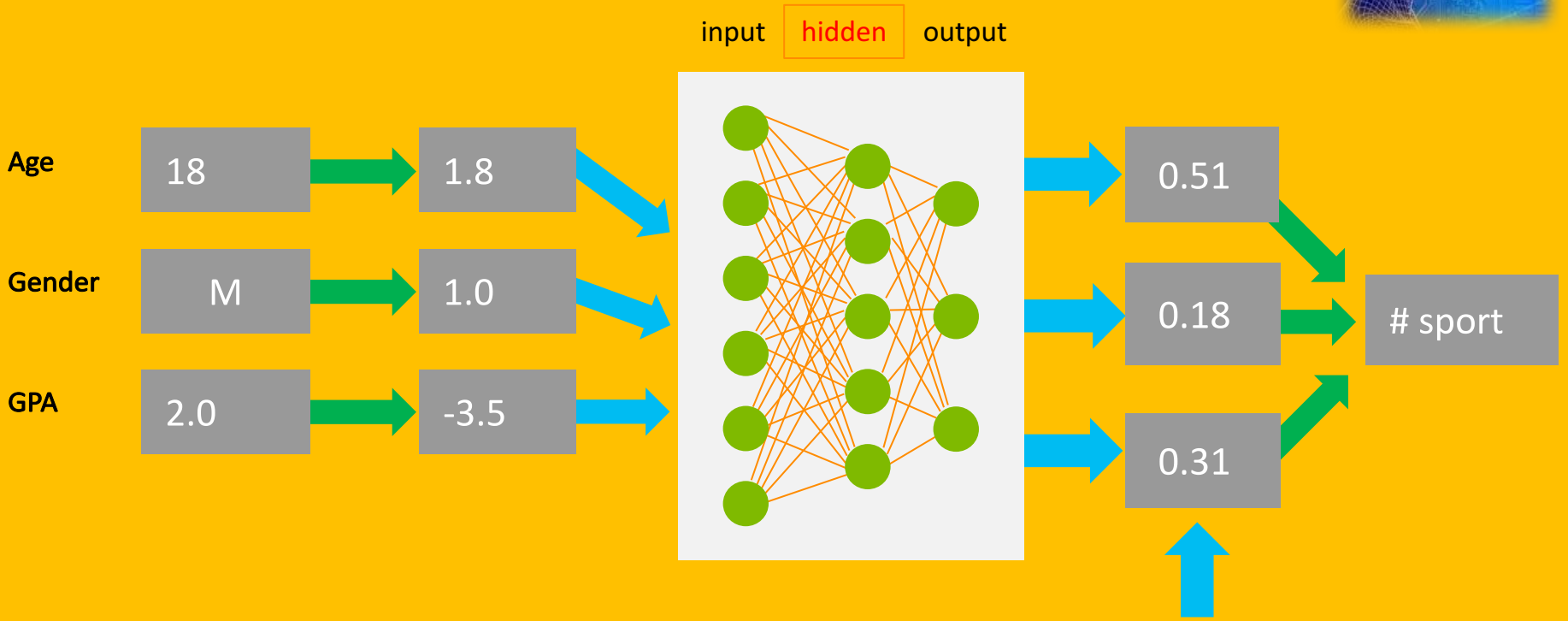
Biological Terminology	**Artificial Neural Network Terminology**
Neuron	Node/Unit/Cell/Neurode
Synapse	Connection/Edge/Link
Synaptic Efficiency	Connection Strength/Weight
Firing frequency	Node output

How does a Neural Network work?

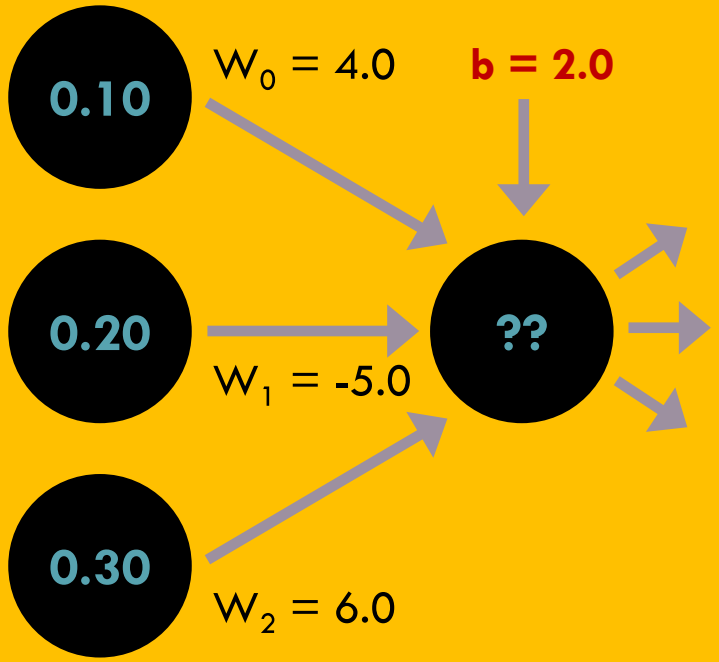
Neural Network Classification



What's going one?



The perceptron



a. (0.1)(4.0) + (0.2)(-5.0) + (0.3)(6.0) = 1.2

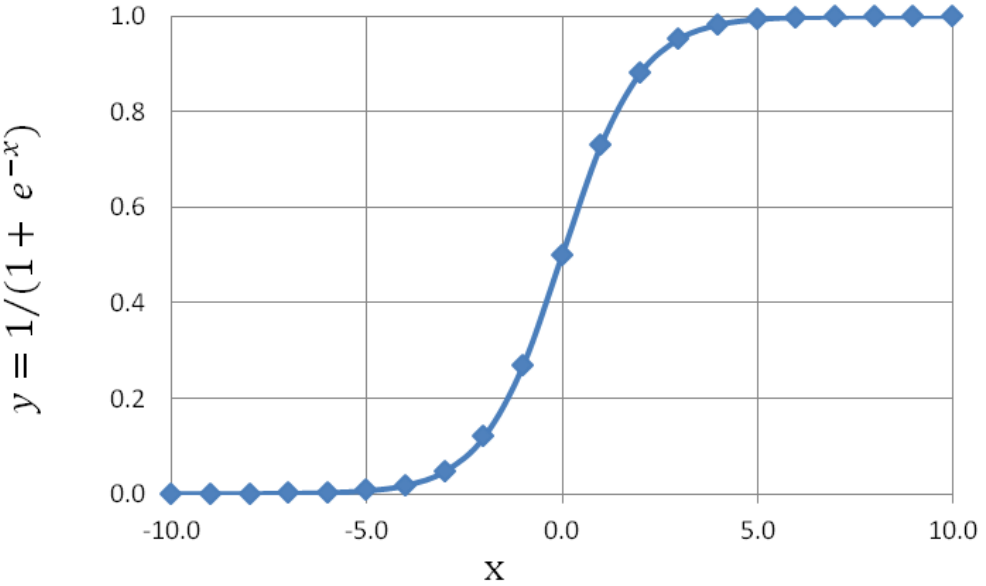
b. 1.2 + 2.0 = 3.2

c. Activation Function (3.2) = 0.73

d. ?? = 0.73

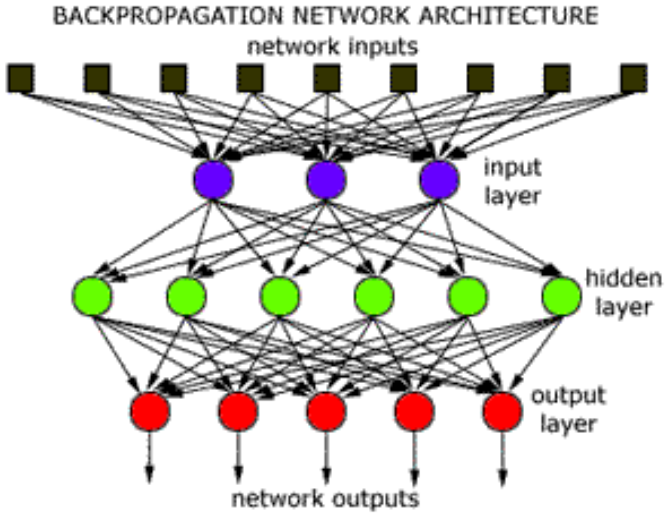
Activation functions

- Logistic sigmoid
  - Output between [0, 1]
  - y = 1.0 / (1.0 + e–x)
- Hyperbolic tangent
  - Output between [-1, +1]
  - y = tanh(x) = (ex – e-x) / (ex + e-x)
- Heaviside step
  - Output either 0 or 1
  - if (x < 0) then y = 0 else if (x >= 0) then y
- Softmax
  - Outputs between [0, 1] and sum to 1.0
  - y = (e-xi) / Σ (e-xj)



Neural Network Training

- Back-propagation
  - Fastest technique.
  - Does not work with Heaviside activation.
  - Requires “learning rate” and “momentum.”
- Genetic algorithm
  - Slowest technique.
  - Generally most effective.
  - Requires “population size,” “mutation rate,” “max generations,” “selection probability.”
- Particle swarm optimization
  - Good compromise.
  - Requires “number particles,” “max iterations,” “cognitive weight,” “social weight.”




SHOW
MOVIECLIPS.COM

Neural Network in code

```
let prepend value (vec:Vector<float>) =
        vector [ yield! value :: (vec |> Vector.toList) ]

let prependForBias : Vector<float> -> Vector<float> = prepend 1.0

let layer activationF (weights:Matrix<float>) (inputs:Vector<float>) =
        (weights * inputs) |> Vector.map activationF  |> prependForBias
```

Neural Network in code

```
let sigmoid x = 1.0 / (1.0 + exp(-x))

let wmd = matrix [ [-1.0; 0.1; 0.3; 0.7];
                   [-2.0; 0.3; 0.2; 1.0];
                   [-1.0; 0.8; 0.6; 0.6];
                   [-2.0; 0.6; 0.3; 0.5]; ]
let wkm = matrix [ [-1.5; 0.5; 0.5; 0.5; 0.5];
                   [-2.5; 0.5; 0.5; 0.5; 0.5]; ]

let twoLayerNetwork : (Vector<float> -> Vector<float>) =
    layer sigmoid wmd >> layer sigmoid wkm

let compute (input : Vector<float>) =
    (twoLayerNetwork <| prependForBias input).[ 1 .. ]
```

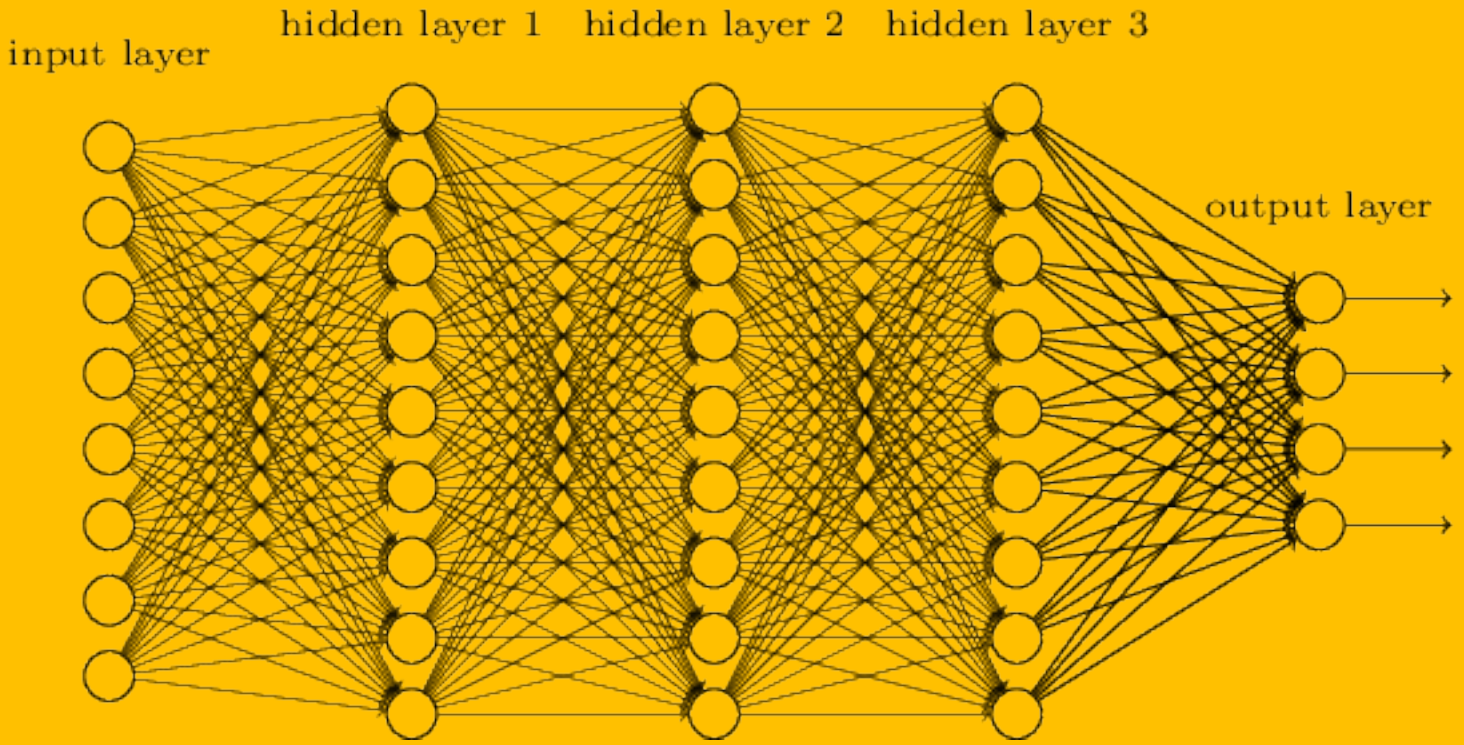
Neural Network in code

```
let out = compute (vector [1.0; 2.0; 3.0])
```

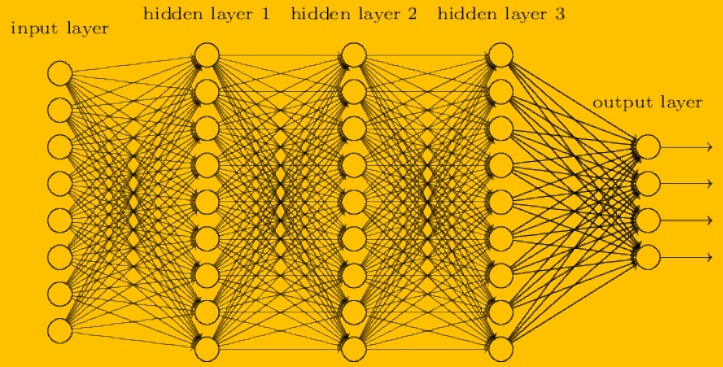
```
RESULT ➔ Vector<float> = seq [0.5392375632; 0.3009608905]
```

Why a Parallel Neural Network?

Why a Parallel Neural Network?



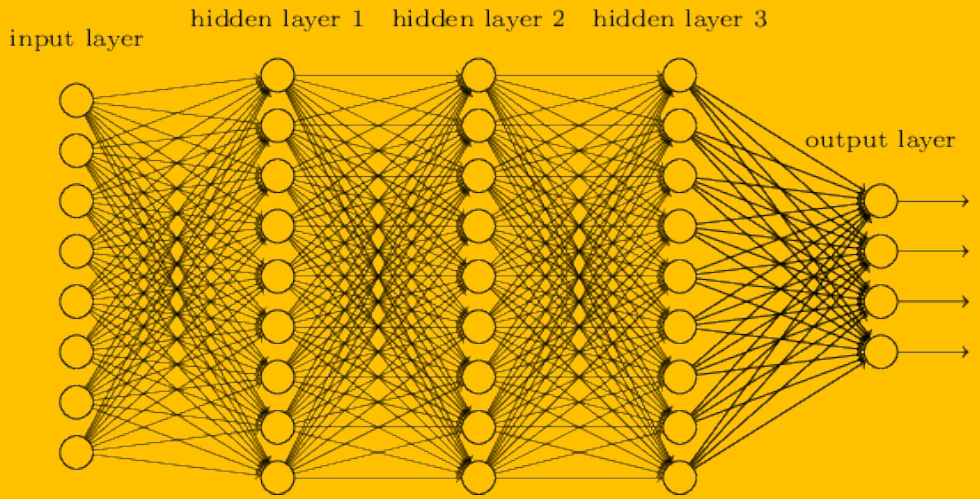
Why a Parallel Neural Network?



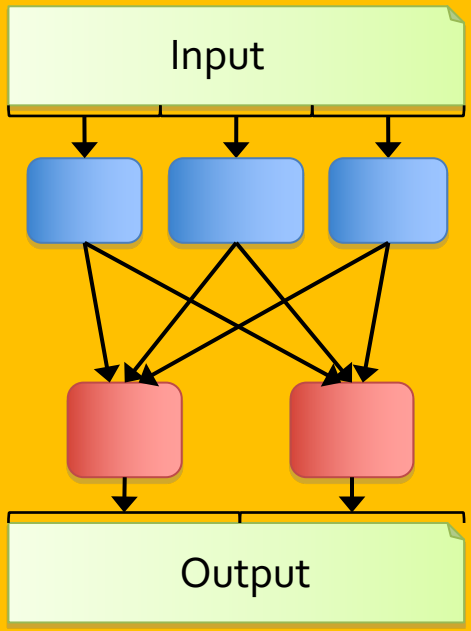
Typical structure of an NN:

- For each training session
  - For each training example in the session
    - For each layer (forward and backward)
      - For each neuron in the layer
        - For all weights of the neuron
          - For all bits of the weight value

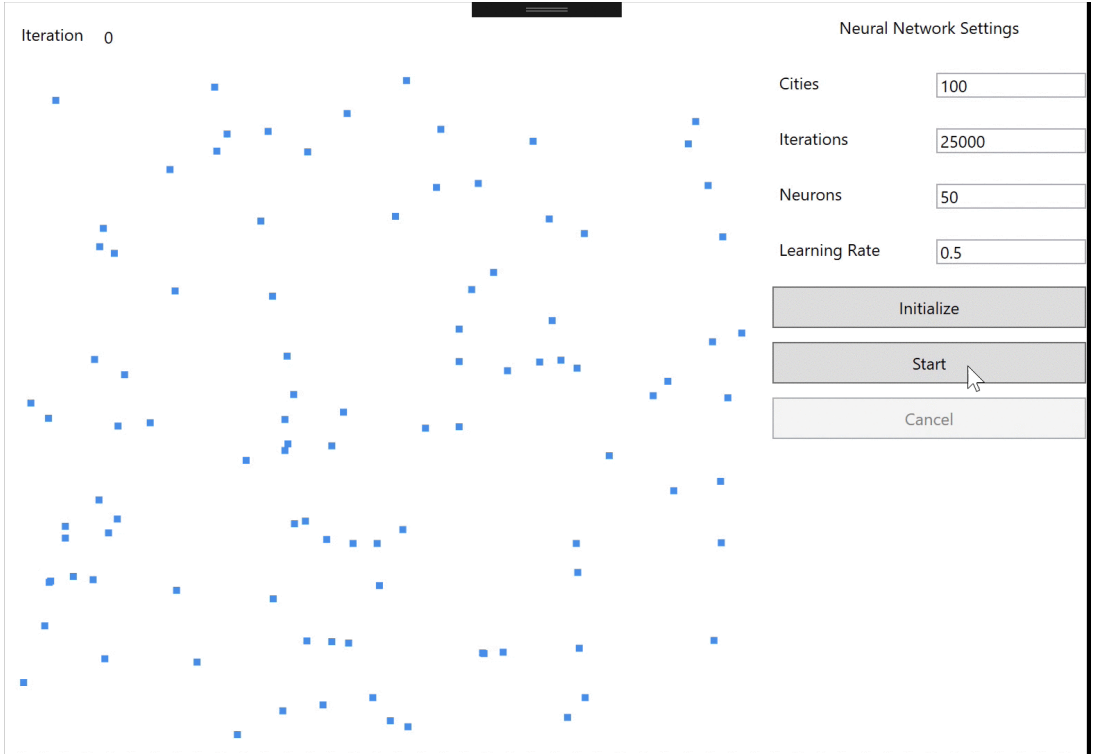
Why a Parallel Neural Network?



MapReduce



The Traveling ~~Salesman~~ Santa Problem



Elastic Network

is a type of neural network which utilizes unsupervised learning algorithms for clusterization problems and treats neural networks as a ring of nodes.

The learning process keeps changing the shape of the ring, where each shape represents a possible solution

The Traveling ~~Salesman~~ Santa Problem

Parallelized neural network system for solving Euclidean traveling salesman problem

Bihter Avşar[a], Danial Esmaeili Aliabadi[a,*]

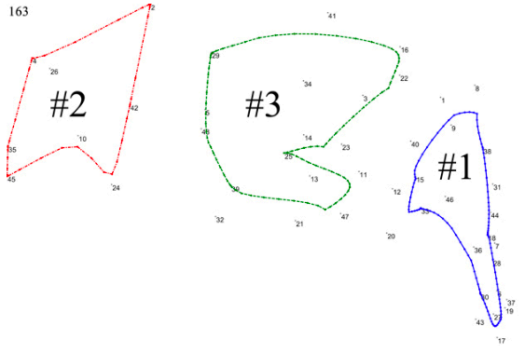
[a] *Sabanci University, Faculty of Engineering and Natural Science, Istanbul, Turkey.*

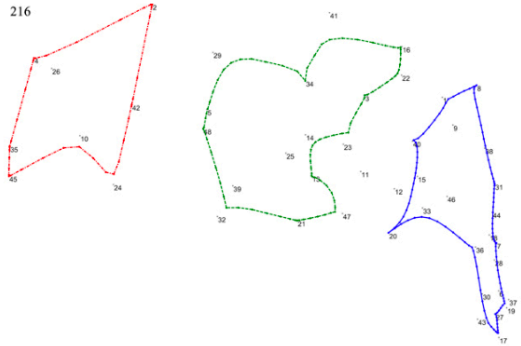
Abstract

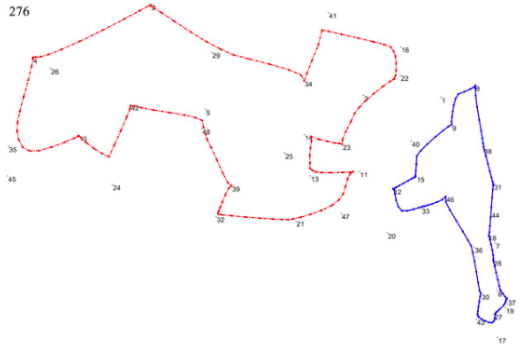
We investigate a parallelized divide-and-conquer approach based on a self-organizing map (SOM) in order to solve the Euclidean Traveling Salesman Problem (TSP). Our approach consists of dividing cities into municipalities, evolving the most appropriate solution from each municipality so as to find the best overall solution and, finally, joining neighborhood municipalities by using a blend operator to identify the final solution. We evaluate performance of parallelized approach over standard TSP test problems (TSPLIB) to show that our approach gives a better answer in terms of quality and time rather than the sequential evolutionary SOM.

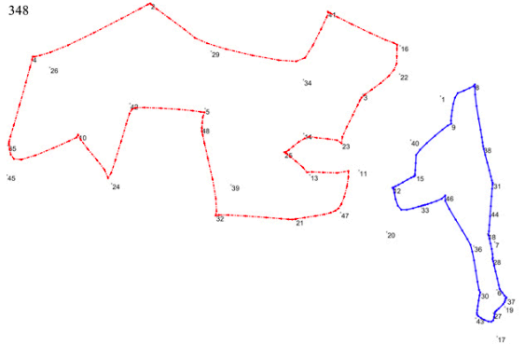
Keywords: Euclidean Traveling Salesman Problem, Artificial Neural Network, Parallelization, Self-organized map, TSPLIB

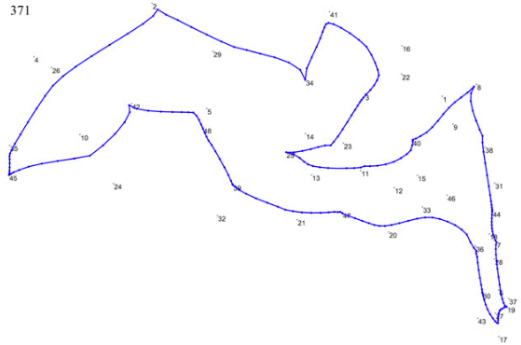
The Traveling ~~Salesman~~ Santa Problem

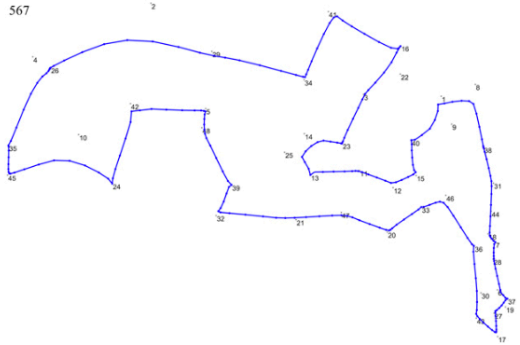


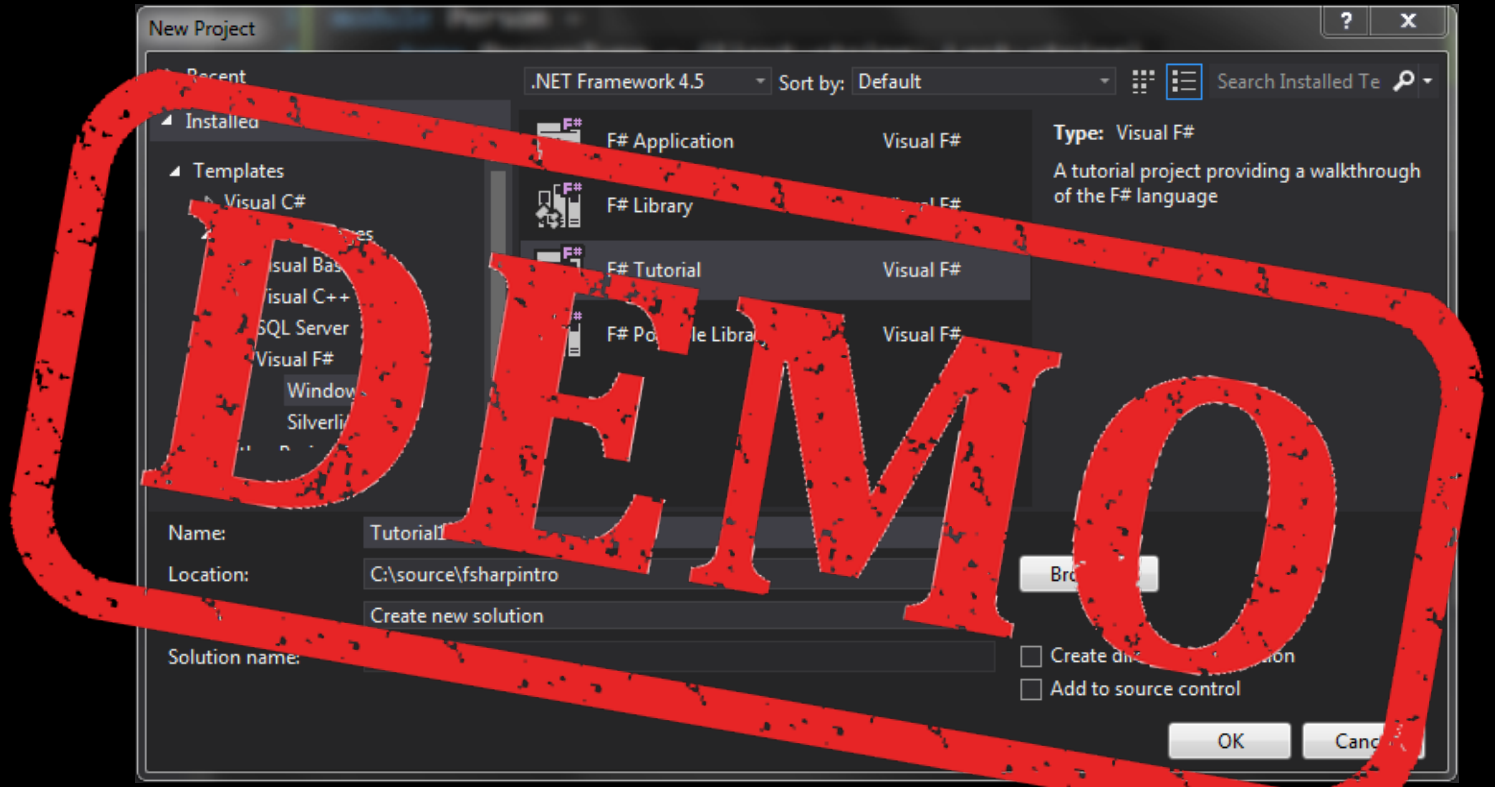











New Project
.NET Framework 4.5
Sort by:
Default
Search Installed Te
Installed
Templates
Visual C#
Visual Basic
Visual C++
SQL Server
Visual F#
Windows
Silverlight
F# Application
Visual F#
F# Library
F# Tutorial
Visual F#
Type: Visual F#
A tutorial project providing a walkthrough of the F# language
Name:
Location:
C:\source\fsharpintro
Create new solution
Solution name:
Add to source control
OK
DEMO

Summary

Neural Networks ease complex problem solving…

Neural Networks are based on simple linear algebra!

Neural Networks can be parallelized!!

…to be very fast

Resources

- ftp://ftp.sas.com/pub/neural/FAQ.html#questions
- Machine Learning Projects for .NET Developers
- Mastering .NET Machine Learning
- Evelina Gabasova blog
- James McCaffrey blog


That's all Folks!

The tools we use have a profound (and devious!) influence on our thinking habits, and, therefore, on our thinking abilities.

-- Edsger Dijkstra

How to reach me

https://github.com/rikace/ml-workshop

meetup.com/DC-fsharp

@DCFsharp @TRikace

tericcardo@gmail.com