Strength-based and time-based supervised networks:

Theory and comparisons

Denis Cousineau

Vision, Integration, Cognition lab.
Université de Montréal

available at: http://mapageweb.umontreal.ca/cousined
Denis.Cousineau@Umontreal.ca

Introduction

- Neural networks are everywhere.
  - Good: the brain is certainly a network of connections between neurons.
  - Bad: Neural networks implement one more assumption than the "network" assumption: The Σ assumption.

- The Σ assumption is a hidden assumption;
  it supposes that all the connections and the inputs are "strength" and that they all contribute to the decision.

- Therefore, standard neural networks should be called:
  Strength-based networks.

- An alternative is to explore the **Time-based networks**:
  - Akin to accumulator models and race models
  - Not another neural net, but a whole family of new neural nets (a new world) based on time.

Summary of the talk

I will draw parallels between **<u>Strength-based</u>** and **<u>Time-based</u>** networks on the following aspects:

- Architecture
- Input-output representations
- Connections
- The mathematics
- A learning rule

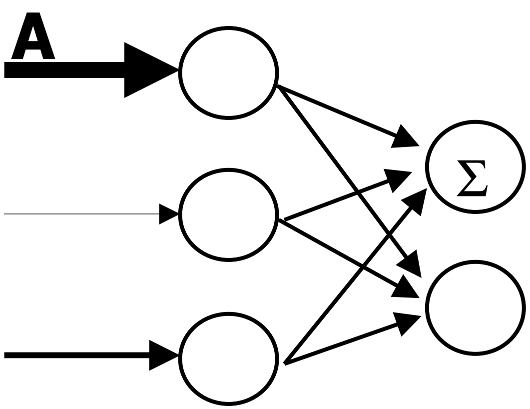
presents tests that show some differences:

- manipulating noise
- manipulating redundancy
- manipulating both

and I may not have time to conclude…

Architecture

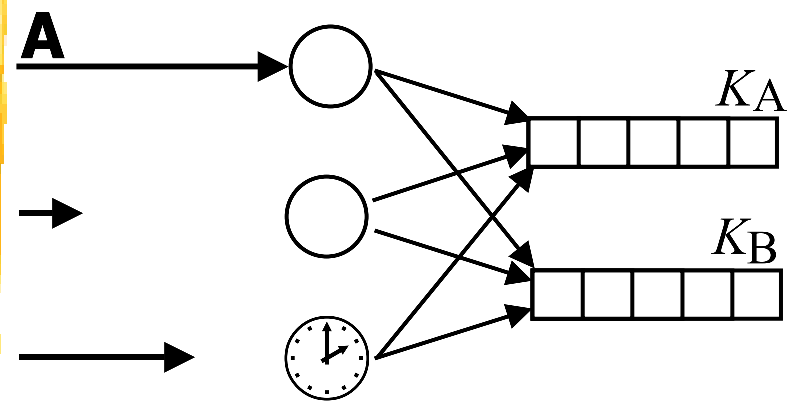
Strength-based



There is some freedom in the architecture:

There can be hidden units which compose an extra layer called the "hidden layer"; they are used to solve non-linear problems, such as the XOR.

Time-based



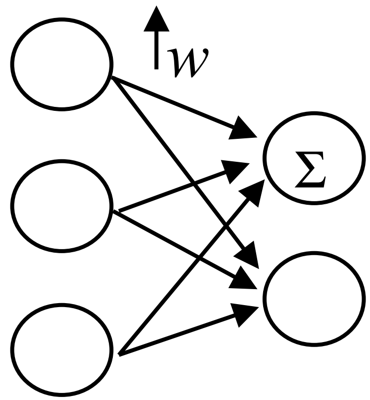
There can be "time-out" units whose action is to become activated after some time; they are needed to make responses when nothing is presented to the network (absence of input).

Input-output representations

Strength-based	Time-based
A, the inputs, are Strength: - importance - saliency	**A**, the inputs, are times: - moment the input is available - saliency
They are either: (on –or– off) (strong –or– weak) (1 –or– 0) and any value in-between	They are either: (there –or– not there) (sudden –or– never) (0 –or– ∞) and any value in-between
O, the outputs, are also strength: Levels of activation of the output units.	**O**, the outputs, are also times: Moments at which the output units becomes activated.
In a distributed representation network, the overall pattern of output is important.	In a race model, the fastest of the output determines the response.
Noise, if present, would be normal.	Noise, if present, would be positive only (e.g. exponential).

Connections

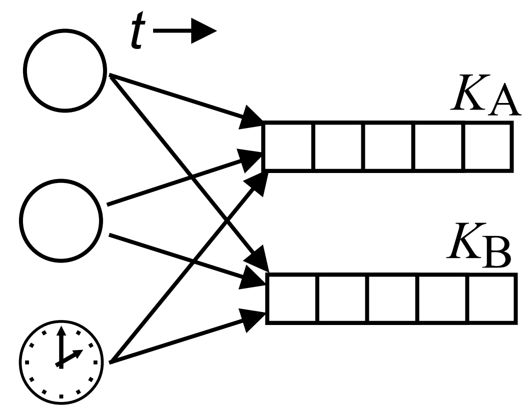
Strength-based



Connections are "weights" that shows how important is the input for the output:

- Relevant input *i* for output *j* should have a high weight ($w_{ij} = 1$);
- Irrelevant input should have no influence on the output ($w_{ij} = 0$).

Time-based



Connections are "delays" that shows how much priority this input has for the output:

- Diagnostic input *i* for a response *j* should fill a slot immediately (no delay, $d_{ij} = 0$);
- Non diagnostic input should never fill a slot ($d_{ij} = \infty$).

The mathematics

Strength-based

The output is the result of an inner product:

$$\mathbf{O} = \mathbf{A}.\mathbf{W} = \mathbf{A} \begin{pmatrix} \times \\ \Sigma \end{pmatrix} \mathbf{W}$$

Standard inner product has a long history, joining pairs of values with $\times$ and aggregating columns with Σ.

Among other properties, it has an "identity matrix" $\mathbf{I}$ such that:

$$\mathbf{I}.\mathbf{A} = \mathbf{A}.\mathbf{I} = \mathbf{A}$$

$$\mathbf{I} = \begin{pmatrix} 1 & 0 & & 0 \\ 0 & 1 & & 0 \\ & & \ldots & \\ 0 & 0 & & 1 \end{pmatrix}$$

Time-based

The output is the result of a "redefined" inner product:

$$\mathbf{O} = \mathbf{A} \tilde{.} \mathbf{D} = \mathbf{A} \begin{pmatrix} + \\ \underset{K}{Min} \end{pmatrix} \mathbf{D}$$

Redefined inner product, noted $\tilde{.}$ has no history, joining pairs of delays with + and aggregating columns by finding the fastests *k* inputs that fill the accumulator.

Surprisingly, it has an identity matrix $\tilde{\mathbf{I}}$ such that:

$$\tilde{\mathbf{I}} \tilde{.} \mathbf{A} = \mathbf{A} \tilde{.} \tilde{\mathbf{I}} = \mathbf{A}$$

$$\tilde{\mathbf{I}} = \begin{pmatrix} 0 & \infty & & \infty \\ \infty & 0 & & \infty \\ & & \ldots & \\ \infty & \infty & & 0 \end{pmatrix}$$

A learning rule

Strength-based	Time-based
The now classic Δ rule: $$\Delta w_{ij} = \alpha \mathbf{A} \ )\times(\ (\mathbf{E} - \mathbf{O})$$	The redefined $\widetilde{\Delta}$ rules: $$\begin{cases} \Delta d_{ij} = \alpha \mathbf{A} \ )+(\ (\mathbf{E} - \mathbf{O}) \\ \Delta \mathbf{K}_j = \beta (\# \mathbf{A} - \mathbf{K}_j) \end{cases}$$
Its purpose is to attribute errors to out-of-phase connections. It uses the standard outer product (noted $)\times($).	Its purpose is to reduce the delays for inputs that were present, and thus might be diagnostic.
It is based on a desired (**E**xpected) output, **E** and is thus a supervised learning.	It is based on a desired output **E** which states at what time the outputs should have been filled; a vector like {t, t, 0, t, t}, t>0.
	The outer product is also redefined: $)+($.

Overview of the simulations

We tested both **strength-based** and **time-based** networks on an identical problem: the XOR problem:

- Activate the first output if none or both of the inputs are on.
- Activate the second output if either one or the other input is on.

The strength-based network had a hidden layer of 4 units; learning rate parameter α was 1.5.

The time-based network had two time-out units; learning rates α was 0.1 and β was 0.5.

We trained the networks for 500 epochs of 10 trials. We computed:

- the RMSE for the strength-based network,
- the P(e) for the time-based network.

We manipulated:

- noise: either none (η = 0), low (η = 5%), or high ($\eta = 10\%$),
- redundancy: either none (ρ = 1), or high (ρ = 8).

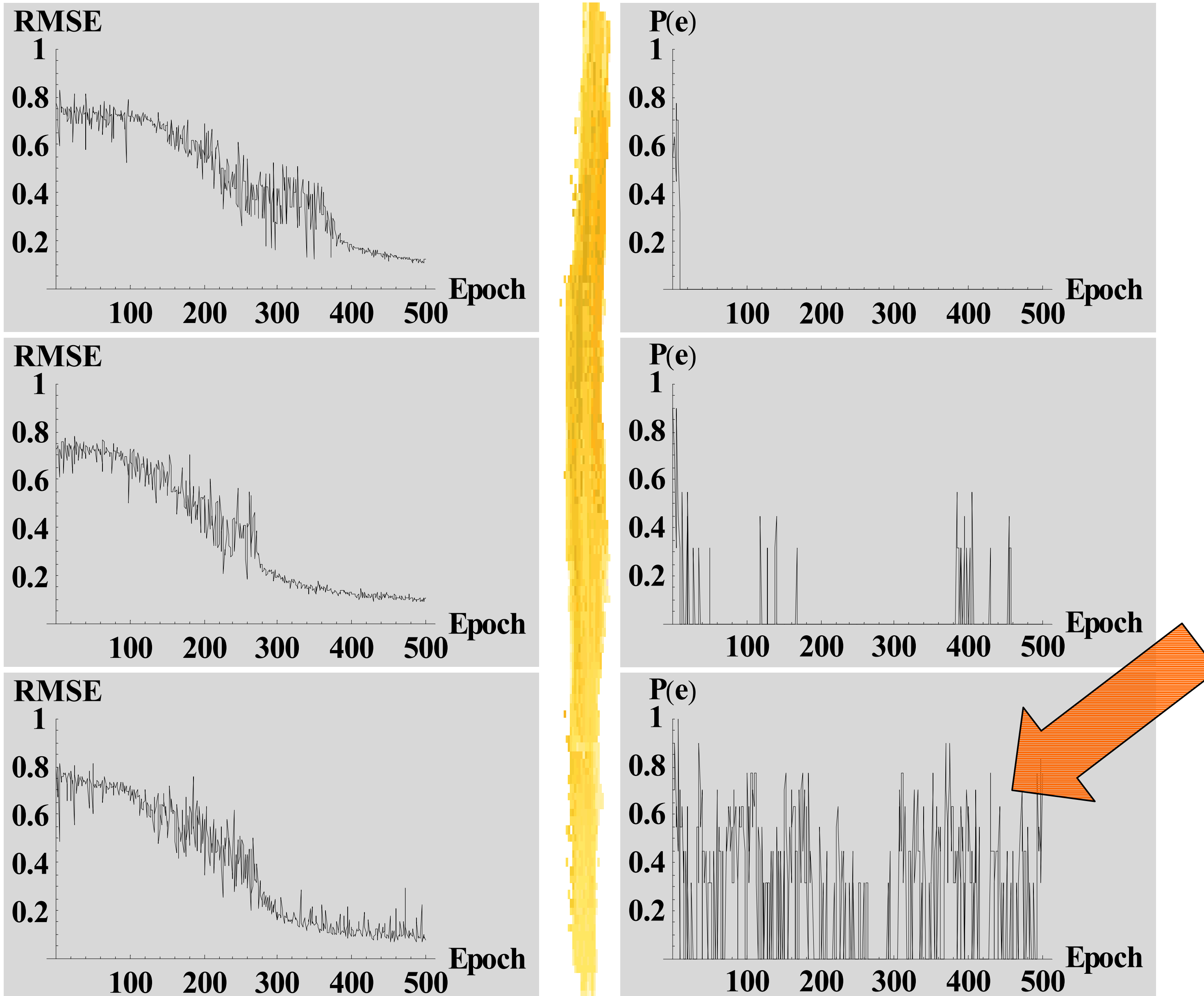
manipulating noise (no redundancy)

Strength-based | **Time-based**

$\eta = 0$

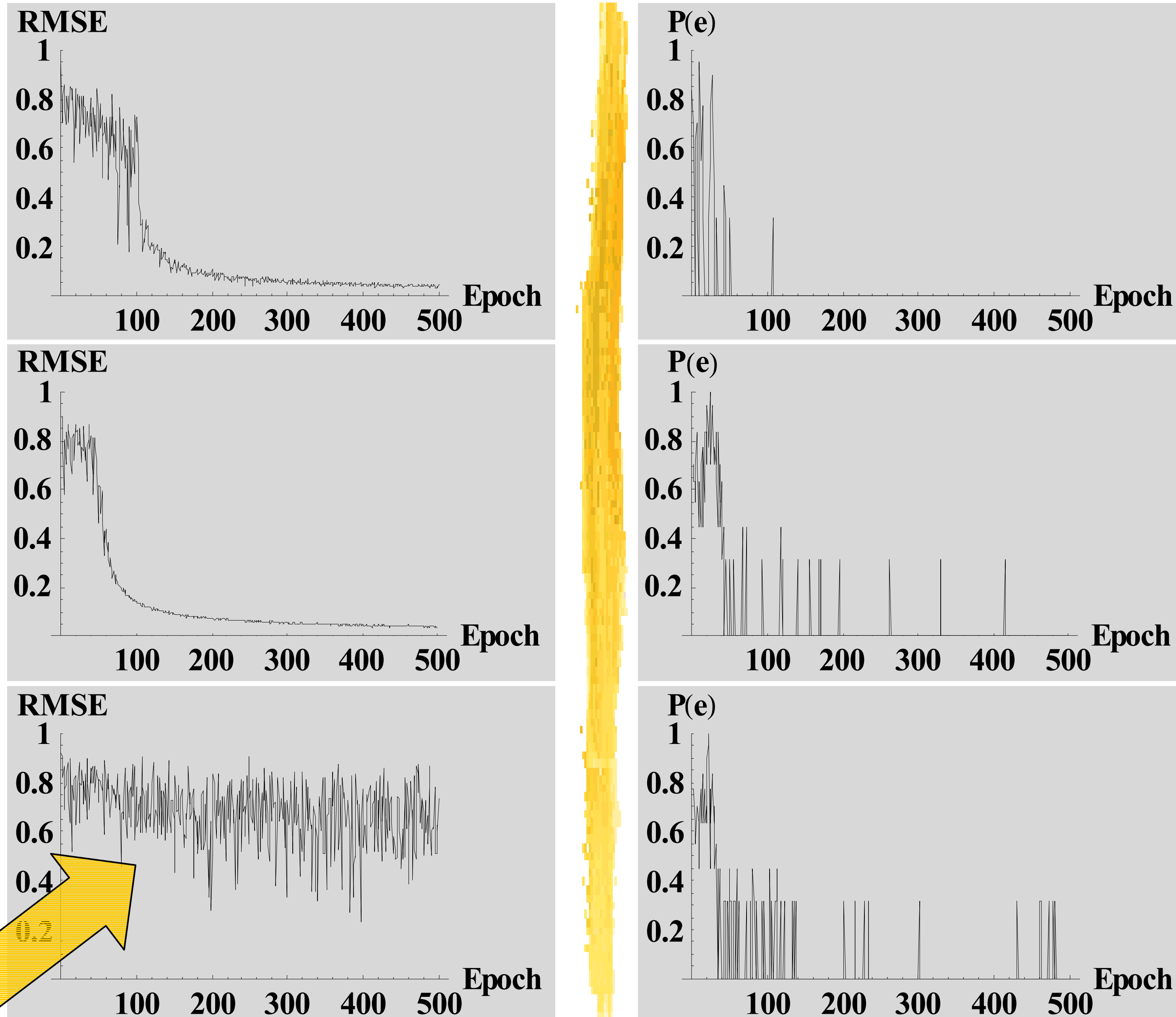
$\eta = 5\%$

$\eta = 10\%$



manipulating noise (high redundancy)

	Strength-based	Time-based
$\eta = 0$		
$\eta = 5\%$		
$\eta = 10\%$		



Conclusions

1- Redundancy:

- Is the nemesis of strength-based networks;
- Is more than likely to be present in the human brain.

2- Time-based networks can predict

- moments (mean $\overline{\mathbf{RT}}$, standard deviation $\overleftrightarrow{\mathbf{RT}}$ and skewness $\nearrow\!\!\!\!\!\mathbf{RT}$),
- speed-accuracy trade-off,
- ROC curves

more efficiently than strength-based networks.

3- The $\tilde{\Delta}$ rule is only one possibility; we will explore:

$$H\tilde{e}bb \qquad S\tilde{O}M$$

4- There is maybe a third family of networks:

- using a multiplicative rule $\mathbf{A}_{\begin{pmatrix}\times\\ \Pi\end{pmatrix}}\mathbf{W}$
- it would be identical to a cascade model, but with a learning rule?