

Corso di Laurea Specialistica in Informatica

# Modelli neurali costruttivi di tipo Reservoir Computing per domini strutturati

Candidato: Giulio Visco

UNIVERSITÀ DI PISA





# Machine Learning e Dati Strutturati

## Machine Learning

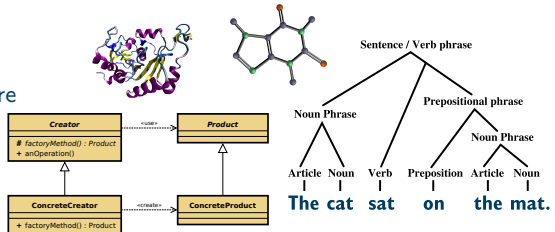
Metodi ed algoritmi per **apprendere da esempi** un compito computazionale definito.

## Machine Learning

Metodi ed algoritmi per **apprendere da esempi** un compito computazionale definito.

Esempi di *dati strutturati* in svariati settori applicativi:

- ▶ Elaborazione del Linguaggio Naturale
- ▶ Ingegneria del Software
- ▶ Chimica
- ▶ Biologia
- ▶ ...molti altri.

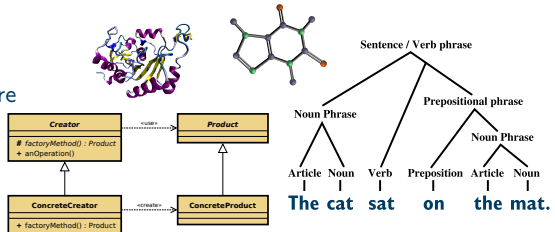


## Machine Learning

Metodi ed algoritmi per **apprendere da esempi** un compito computazionale definito.

Esempi di *dati strutturati* in svariati settori applicativi:

- Elaborazione del Linguaggio Naturale
- Ingegneria del Software
- Chimica
- Biologia
- ...molti altri.



## Obiettivo

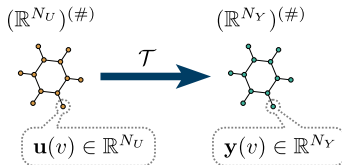
*Estendere le metodologie* di Machine Learning per il trattamento di *Domini Strutturati*.



# Apprendimento su Domini Strutturati

## Approccio Neurale

Apprendere  
*trasduzioni strutturali*:





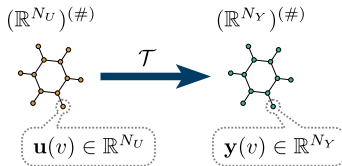
# Apprendimento su Domini Strutturati

## Approccio Neurale

Apprendere  
*trasduzioni strutturali*:

Vari approcci possibili:

kernel, ILP, approcci generativi e probabilistici, relational learning...

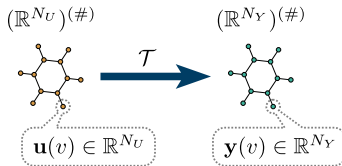




# Apprendimento su Domini Strutturati

## Approccio Neurale

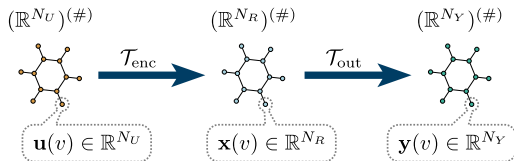
Apprendere  
*trasduzioni strutturali*:



Vari approcci possibili:

kernel, ILP, approcci generativi e probabilistici, relational learning...

...*Reti Neurali Ricorsive*:  $\mathcal{T} = \mathcal{T}_{\text{out}} \circ \mathcal{T}_{\text{enc}}$

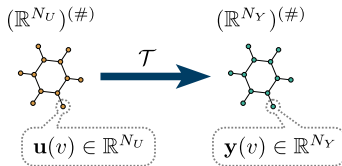




# Apprendimento su Domini Strutturati

## Approccio Neurale

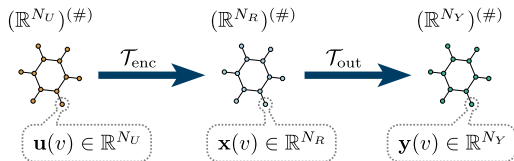
Apprendere  
*trasduzioni strutturali*:



Vari approcci possibili:

kernel, ILP, approcci generativi e probabilistici, relational learning...

...*Reti Neurali Ricorsive*:  $\tau = \tau_{\text{out}} \circ \tau_{\text{enc}}$



Apprendimento

Tipicamente *tutti* i pesi della rete vengono addestrati.





# Reti Neurali Ricorsive

## Limiti

Nella loro formulazione classica presentano dei limiti:



# Reti Neurali Ricorsive

## Limiti

Nella loro formulazione classica presentano dei limiti:

### Input

Per garantire la convergenza del processo di encoding, gli input trattabili devono presentare un ordinamento topologico fra i vertici (i.e. *Grafì Diretti Aciclici*).



# Reti Neurali Ricorsive

## Limiti

Nella loro formulazione classica presentano dei limiti:

### Input

Per garantire la convergenza del processo di encoding, gli input trattabili devono presentare un ordinamento topologico fra i vertici (i.e. *Grafi Diretti Aciclici*).

### Costo

*Costo computazionale elevato* dell'apprendimento.



# Reti Neurali Ricorsive

## Limiti

Nella loro formulazione classica presentano dei limiti:

### Input

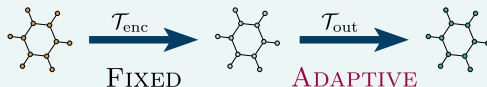
Per garantire la convergenza del processo di encoding, gli input trattabili devono presentare un ordinamento topologico fra i vertici (i.e. *Grafi Diretti Aciclici*).

### Costo

*Costo computazionale elevato* dell'apprendimento.

### Reservoir Computing: Idea

Perché allenare l'intera rete e non solo una parte?





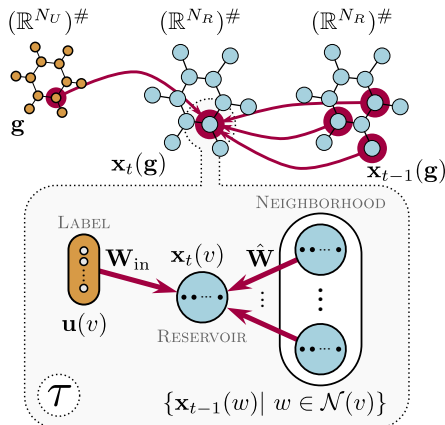
# GraphESN (2010)

Modello di *Reservoir Computing* per l'apprendimento su  
*Domini Strutturati*.

## Modello di *Reservoir Computing* per l'apprendimento su *Domini Strutturati*.

### Reservoir:

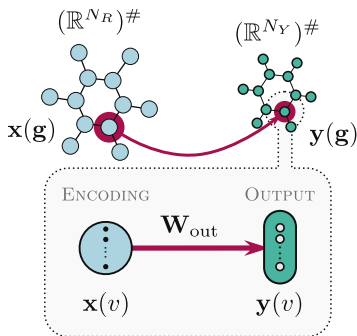
- Implementa un **sistema dinamico** complesso.
- Realizza l'**encoding**,  $\mathcal{T}_{\text{enc}}$ , in maniera dipendente dalla topologia dell'input.
- **Convergenza garantita** dalla contrattività.



## Modello di *Reservoir Computing* per l'apprendimento su *Domini Strutturati*.

### Reservoir:

- Implementa un **sistema dinamico** complesso.
- Realizza l'**encoding**,  $\mathcal{T}_{\text{enc}}$ , in maniera dipendente dalla topologia dell'input.
- **Convergenza garantita** dalla contrattività.



### Readout:

- Una **funzione lineare** che realizza  $\mathcal{T}_{\text{out}}$ .
- **È l'unica parte allenata della rete!**



### Pro

Le proprietà del reservoir garantiscono la convergenza dell'encoding:

- ▶ Si possono apprendere *trasduzioni definite su grafi*.
- ▶ Si possono usare *algoritmi di apprendimento semplici ed efficienti* per il solo readout.



## Pro

Le proprietà del reservoir garantiscono la convergenza dell'encoding:

- ▶ Si possono apprendere *trasduzioni definite su grafi*.
- ▶ Si possono usare *algoritmi di apprendimento semplici ed efficienti* per il solo readout.

## Contro

I reservoir sono fissati a priori:

- ▶ Difficile *determinare le dimensioni corrette* per il problema.
- ▶ Realizzano un processo di *encoding non adattivo*.



# Estendere GraphESN

## Strategia Costruttiva

### Obiettivo

Sfruttare le caratteristiche delle GraphESN:

*mantenerne i vantaggi e affrontarne i limiti.*



# Estendere GraphESN

## Strategia Costruttiva

### Obiettivo

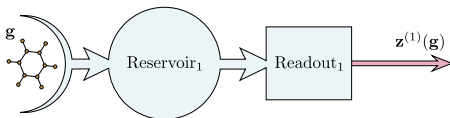
Sfruttare le caratteristiche delle GraphESN:

*mantenerne i vantaggi e affrontarne i limiti.*

### Come fare?

Introduzione dell'*approccio costruttivo*: realizzare la rete attraverso un procedimento iterativo, usando più *sotto-reti*.

(i) Allenamento della prima sotto-rete.

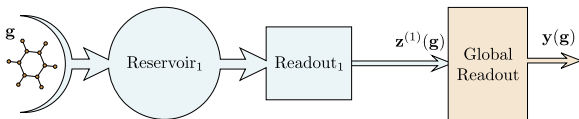




# GraphESN Constructive Flat

## Costruzione della Rete

(ii) Collegamento della sotto-rete ad un readout globale.

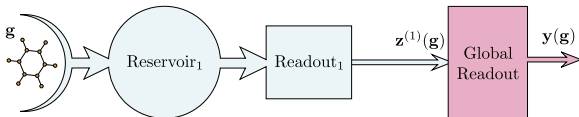




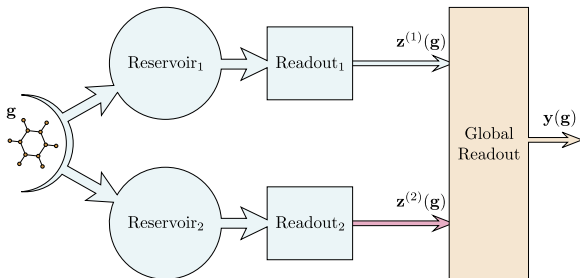
# GraphESN Constructive Flat

## Costruzione della Rete

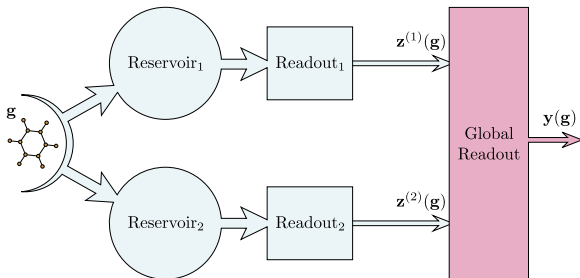
(iii) Allenamento del readout globale. Si ricava l'errore commesso.



(iv) Si allena la seconda sotto-rete per emulare l'errore commesso.

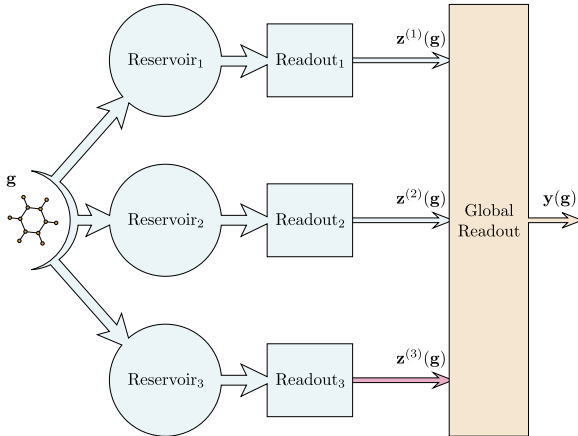


(v) Si allena il readout globale sfruttando il segnale di correzione.

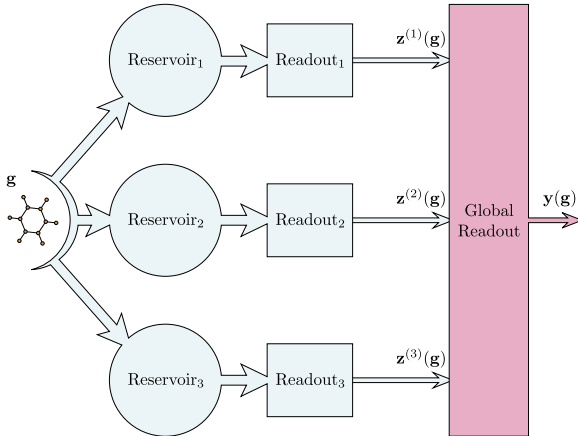




(vi) Si aggiunge una sotto-rete per correggere il nuovo errore.



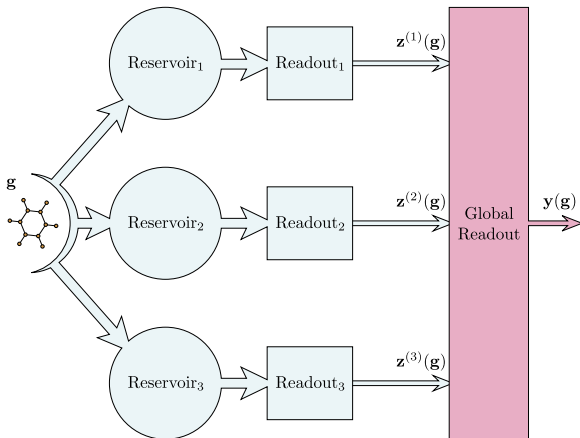
(...) Si procede finché l'errore non sia sufficientemente basso.



# GraphESN Constructive Flat

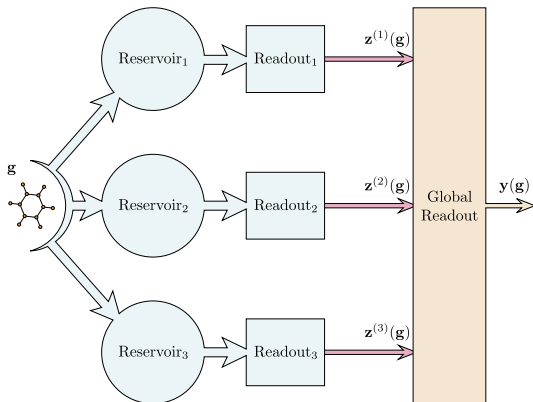
## Costruzione della Rete

La *dimensione* della rete dipende direttamente dal *problema*.



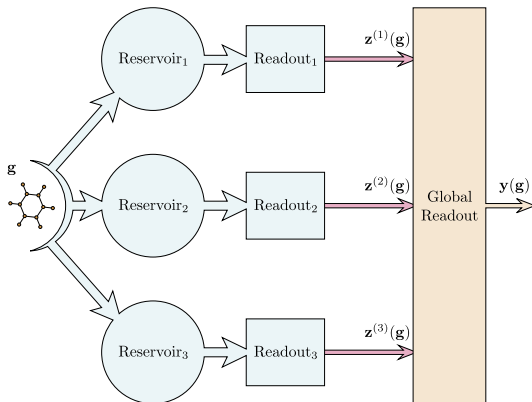
# GraphESN Constructive Flat (GraphESN-CF)

- Scomporre un problema in **sotto-problemi**.



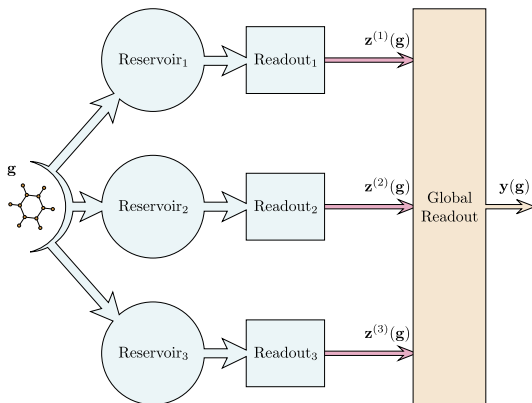
# GraphESN Constructive Flat (GraphESN-CF)

- Scomporre un problema in **sotto-problemi**.
- Utilizzare **unità** computazionali più **semplici**.



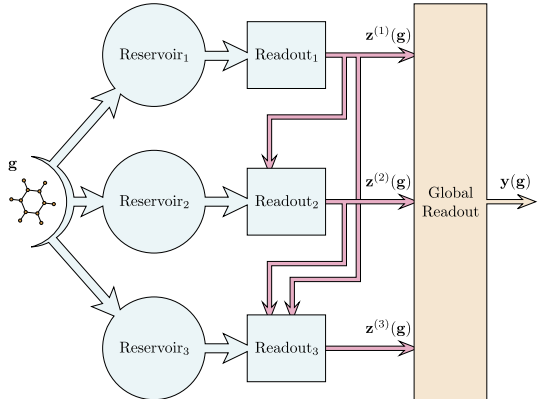
# GraphESN Constructive Flat (GraphESN-CF)

- Scomporre un problema in **sotto-problemi**.
- Utilizzare **unità** computazionali più **semplici**.
- Determinare **automaticamente** la **topologia**.



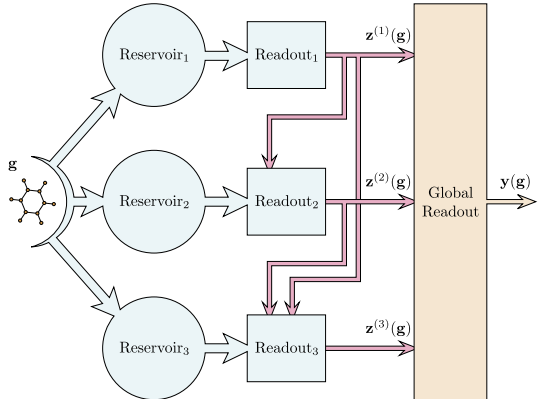
# GraphESN Forward (GraphESN-FW)

- Connessioni **verso i readout** delle sotto-reti successive.



# GraphESN Forward (GraphESN-FW)

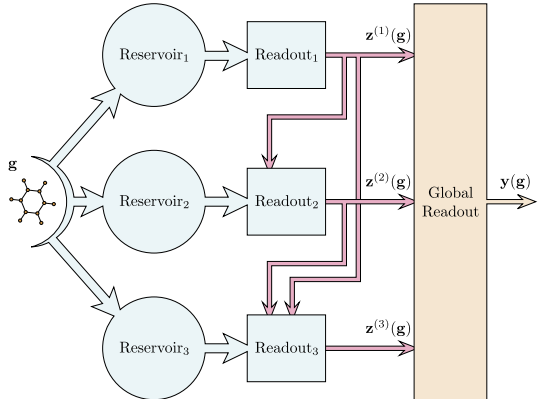
- Connessioni **verso i readout** delle sotto-reti successive.
- Sfruttare **informazioni già apprese** durante il processo iterativo.





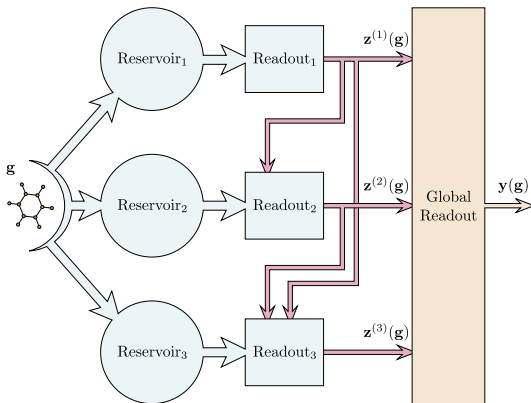
# GraphESN Forward (GraphESN-FW)

- Connessioni **verso i readout** delle sotto-reti successive.
- Sfruttare **informazioni già apprese** durante il processo iterativo.
- **Semplificare** la risoluzione dei sotto-problemi.



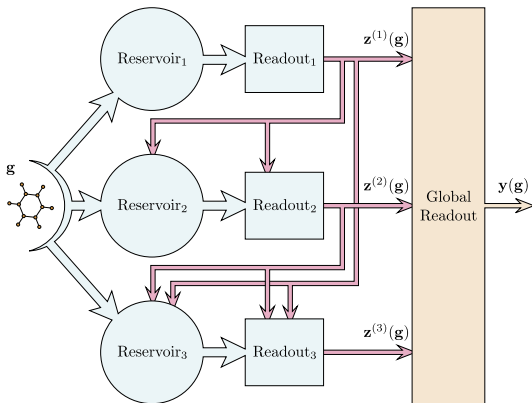
# GraphESN Forward (GraphESN-FW)

- Connessioni **verso i readout** delle sotto-reti successive.
- Sfruttare **informazioni già apprese** durante il processo iterativo.
- **Semplificare** la risoluzione dei sotto-problemi.
- **Multilayer non lineare** di tipo Reservoir Computing.



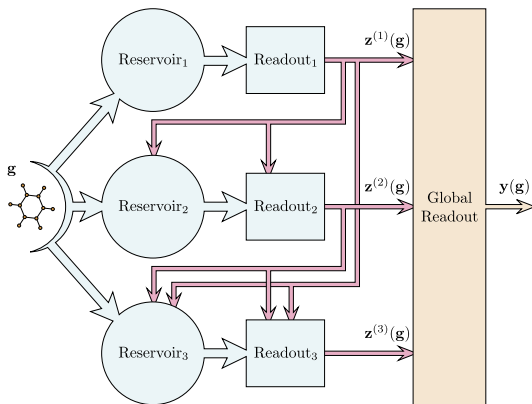
# GraphESN Forward Output-Feedback (GraphESN-FOF)

- Connessioni anche **verso i reservoir** delle sotto-reti successive. (**output-feedback**)



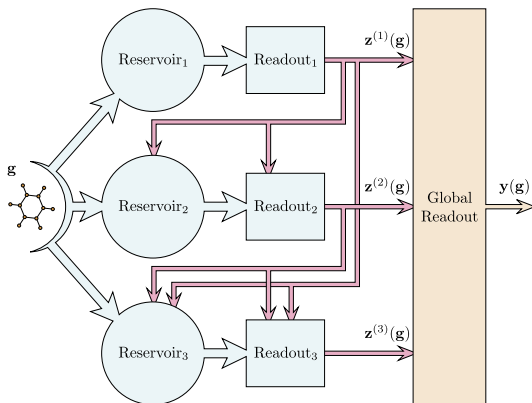
# GraphESN Forward Output-Feedback (GraphESN-FOF)

- Connessioni anche **verso i reservoir** delle sotto-reti successive. (**output-feedback**)
- **Output-feedback** per introdurre **informazioni supervisionate** nei reservoir.



# GraphESN Forward Output-Feedback (GraphESN-FOF)

- Connessioni anche **verso i reservoir** delle sotto-reti successive. (**output-feedback**)
- **Output-feedback** per introdurre **informazioni supervisionate** nei reservoir.
- Processo di **encoding** non adattivo ma **guidato dal problema** nonostante i reservoir fissi.





# Vantaggio Computazionale

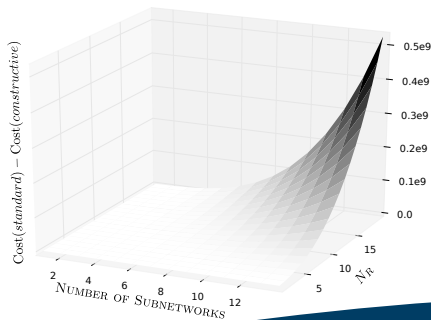
## Topologia Automatica

**Costo GraphESN-FOF:** dim. reservoir  $N_R$ , num. sotto-reti  $NSN$ , dim. dataset  $|\mathcal{G}|$ .

$$O(NSN^4 + N_R NSN^3 + N_R^2 NSN^2 + N_R^3 NSN \\ + NSN^3 |\mathcal{G}| + NSN^2 N_R |\mathcal{G}| + NSN N_R^2 |\mathcal{G}|)$$

**Costo GraphESN (model selection):**  $N'_R \in \{N_R, 2N_R, \dots, NSN N_R\}$

$$O(N_R^3 NSN^4 + N_R^2 NSN^3 |\mathcal{G}|)$$



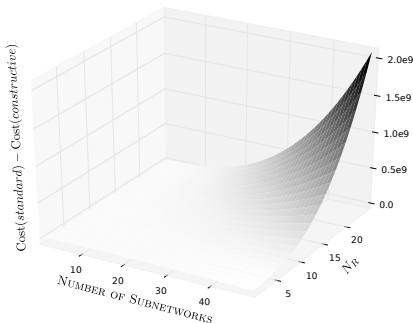


# Vantaggio Computazionale

## Scomposizione del Problema

**Costo GraphESN equivalente:** dimensione reservoir  $N'_R = N_R NSN$ .

$$O(N_R^3 NSN^3 + N_R^2 NSN^2 |\mathcal{G}|)$$



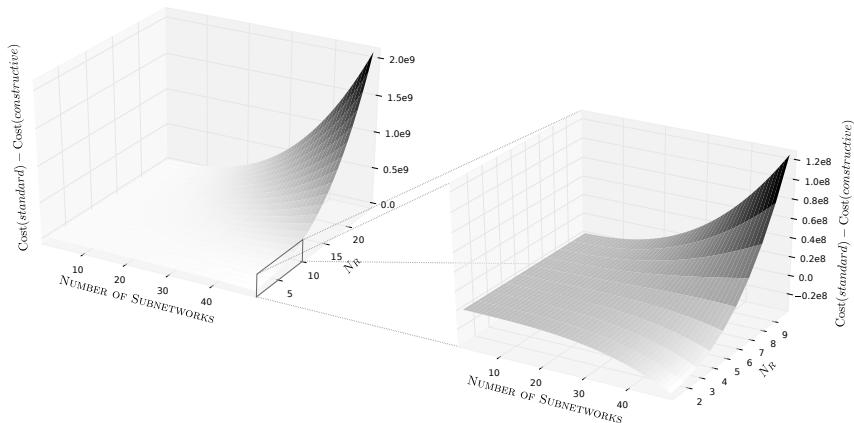


# Vantaggio Computazionale

## Scomposizione del Problema

**Costo GraphESN equivalente:** dimensione reservoir  $N'_R = N_R NSN$ .

$$O(N_R^3 NSN^3 + N_R^2 NSN^2 |\mathcal{G}|)$$







# Analisi Sperimentale Dataset

Dataset reali nell'ambito della Chemioinformatica. **9 task** distinti.



# Analisi Sperimentale Dataset

Dataset reali nell'ambito della Chemioinformatica. **9 task** distinti.

- ▶ **Predictive Toxicology Challenge (PTC):**

- ▶ Numero molecole: 417. Dimensione delle etichette di input: 24.
- ▶ Classificazione dei composti tossici (riferito a **4 tipi** di roditori).

- ▶ **Mutagenesis (Mutag):**

- ▶ Numero molecole: 230. Dimensione etichette di input: 11, 13, 15.
- ▶ Classificazione dei composti mutageni (**3 tipi** di rappresentazione).

- ▶ **Bursi:**

- ▶ Numero molecole: 4204. Dimensione delle etichette di input: 14.
- ▶ Classificazione dei composti mutageni.

- ▶ **Angiotensin Converting Enzyme (ACE):**

- ▶ Numero molecole: 114. Dimensione delle etichette di input: 8.
- ▶ Predizione dell'attività molecolare (i.e.  $pIC_{50}$ ) su intervallo continuo.



# Analisi Sperimentale Dataset

Dataset reali nell'ambito della Chemioinformatica. **9 task** distinti.

- ▶ **Predictive Toxicology Challenge (PTC):**

- ▶ Numero molecole: 417. Dimensione delle etichette di input: 24.
- ▶ Classificazione dei composti tossici (riferito a **4 tipi** di roditori).

- ▶ **Mutagenesis (Mutag):**

- ▶ Numero molecole: 230. Dimensione etichette di input: 11, 13, 15.
- ▶ Classificazione dei composti mutageni (**3 tipi** di rappresentazione).

- ▶ **Bursi:**

- ▶ Numero molecole: 4204. Dimensione delle etichette di input: 14.
- ▶ Classificazione dei composti mutageni.

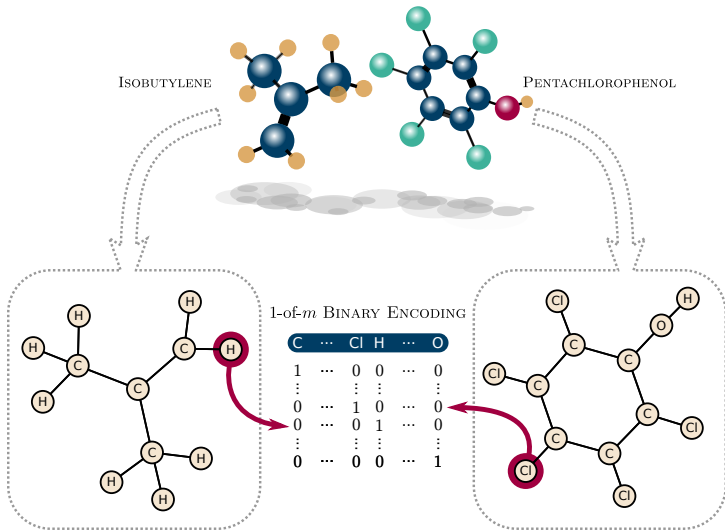
- ▶ **Angiotensin Converting Enzyme (ACE):**

- ▶ Numero molecole: 114. Dimensione delle etichette di input: 8.
- ▶ Predizione dell'attività molecolare (i.e.  $pIC_{50}$ ) su intervallo continuo.

Model selection e validazione con *double cross-validation*.

# Analisi Sperimentale

## Rappresentazione dei Dati





# Analisi Sperimentale

## Performance: PTC - Mutag

### PTC

Model	FR	FM	MR	MM
GraphESN	67.7 ( $\pm 0.1$ )	60.7 ( $\pm 0.4$ )	56.7 ( $\pm 0.9$ )	67.1 ( $\pm 0.1$ )
GraphESN-CF	67.4 ( $\pm 0.8$ )	<b>63.0</b> ( $\pm 0.8$ )	<b>57.9</b> ( $\pm 0.6$ )	65.2 ( $\pm 0.6$ )
GraphESN-FW	67.4 ( $\pm 1.0$ )	<b>63.6</b> ( $\pm 0.6$ )	<b>58.3</b> ( $\pm 0.7$ )	64.3 ( $\pm 1.5$ )
GraphESN-FOF	<b>68.1</b> ( $\pm 1.2$ )	<b>62.7</b> ( $\pm 1.4$ )	<b>57.1</b> ( $\pm 1.3$ )	66.6 ( $\pm 1.7$ )

### Mutag

Model	AB	AB+C	AB+C+PS
GraphESN	75.2 ( $\pm 0.8$ )	76.5 ( $\pm 0.8$ )	80.3 ( $\pm 0.8$ )
GraphESN-CF	<b>79.6</b> ( $\pm 2.5$ )	76.0 ( $\pm 1.5$ )	79.7 ( $\pm 0.2$ )
GraphESN-FW	<b>79.7</b> ( $\pm 2.8$ )	<b>76.7</b> ( $\pm 2.5$ )	<b>80.6</b> ( $\pm 1.9$ )
GraphESN-FOF	<b>76.8</b> ( $\pm 3.7$ )	<b>77.0</b> ( $\pm 2.4$ )	80.0 ( $\pm 2.5$ )



# Analisi Sperimentale

Performance: Bursi - ACE

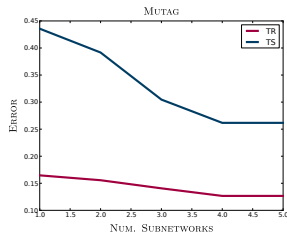
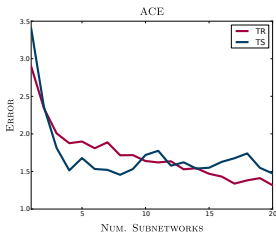
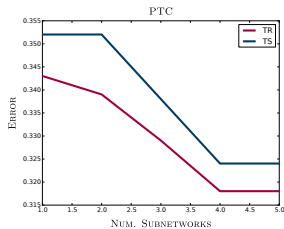
## Bursi

Model	TR	TS
GraphESN	77.9 ( $\pm 0.2$ )	76.2 ( $\pm 0.2$ )
GraphESN-CF	<b>78.3</b> ( $\pm 0.4$ )	<b>76.7</b> ( $\pm 0.7$ )
GraphESN-FW	<b>78.4</b> ( $\pm 0.8$ )	<b>77.1</b> ( $\pm 0.7$ )
GraphESN-FOF	<b>79.8</b> ( $\pm 0.5$ )	<b>78.0</b> ( $\pm 0.9$ )

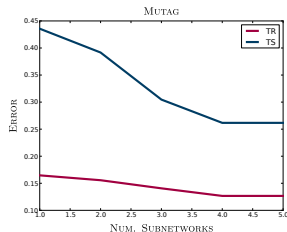
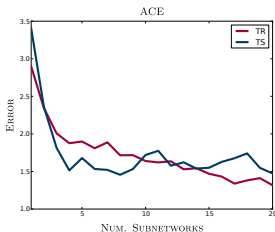
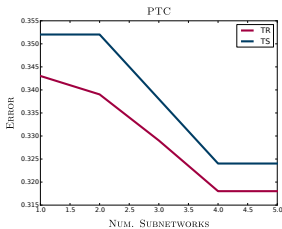
## ACE

Model	MSE (TR)	MSE (TS)
GraphESN	1.83 ( $\pm 0.01$ )	2.13 ( $\pm 0.03$ )
GraphESN-CF	2.03 ( $\pm 0.07$ )	2.37 ( $\pm 0.15$ )
GraphESN-FW	<b>1.71</b> ( $\pm 0.02$ )	<b>2.02</b> ( $\pm 0.06$ )
GraphESN-FOF	<b>1.60</b> ( $\pm 0.07$ )	<b>2.03</b> ( $\pm 0.06$ )

Errore di training (TR) e di test (TS) al crescere della rete.



Errore di training (TR) e di test (TS) al crescere della rete.



Nel corso del processo di costruzione della rete, la *capacità di predizione* media *aumenta*, anche grazie all'applicazione di *tecniche di regolarizzazione*.



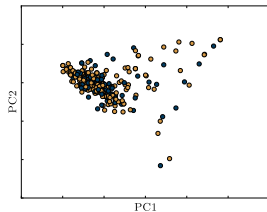
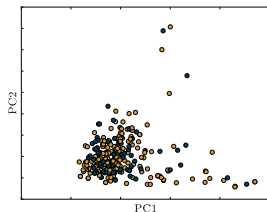


# Analisi Sperimentale

## Principal Component Analysis

Prime due componenti principali dell'*encoding* degli input.

Con output-feedback



Senza output-feedback

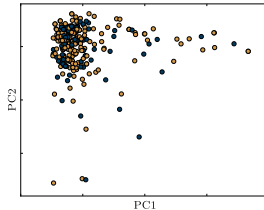
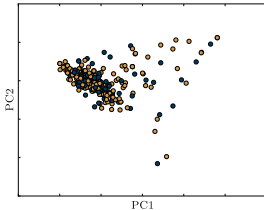
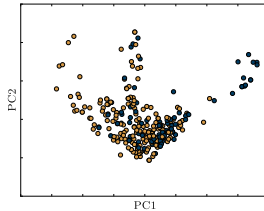
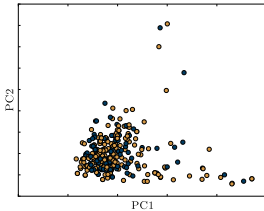


# Analisi Sperimentale

## Principal Component Analysis

Prime due componenti principali dell'*encoding* degli input.

Con output-feedback



Senza output-feedback

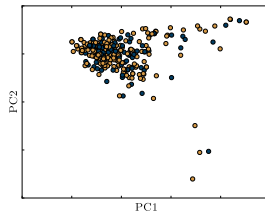
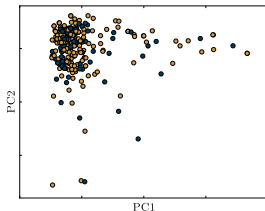
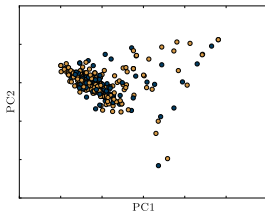
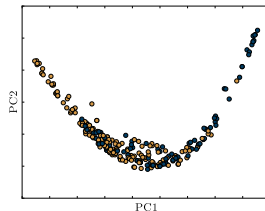
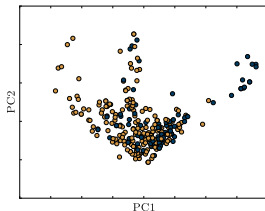
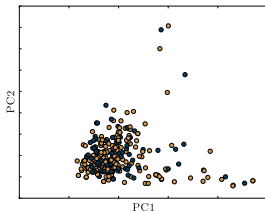


# Analisi Sperimentale

## Principal Component Analysis

Prime due componenti principali dell'*encoding* degli input.

Con output-feedback



Senza output-feedback



# Conclusioni

**Approccio costruttivo nell'ambito del Reservoir Computing:**



# Conclusioni

**Approccio costruttivo** nell'ambito del **Reservoir Computing**:

## Reservoir Computing

- ▶ Apprendere trasduzioni strutturali su grafi.
- ▶ Algoritmi di apprendimento efficienti.



# Conclusioni

## Approccio costruttivo nell'ambito del Reservoir Computing:

### Reservoir Computing

- ▶ Apprendere trasduzioni strutturali su grafi.
- ▶ Algoritmi di apprendimento efficienti.

### Strategia costruttiva

- ▶ Determinazione automatica dell'architettura, guidata dal problema.
- ▶ Vantaggio computazionale senza perdita di accuratezza.
- ▶ Strategia flessibile.



# Conclusioni

## Approccio costruttivo nell'ambito del Reservoir Computing:

### Reservoir Computing

- ▶ Apprendere trasduzioni strutturali su grafi.
- ▶ Algoritmi di apprendimento efficienti.

### Strategia costruttiva

- ▶ Determinazione automatica dell'architettura, guidata dal problema.
- ▶ Vantaggio computazionale senza perdita di accuratezza.
- ▶ Strategia flessibile.

### Output-feedback

- ▶ Informazione supervisionata nel processo di encoding.
- ▶ Non comporta l'allenamento del reservoir.



Claudio Gallicchio, Alessio Micheli, e Giulio Visco. Constructive reservoir computation with output feedbacks for structured domains. In *Proceedings of the ESANN 2012*, 2012. To appear.

*Grazie dell'attenzione*