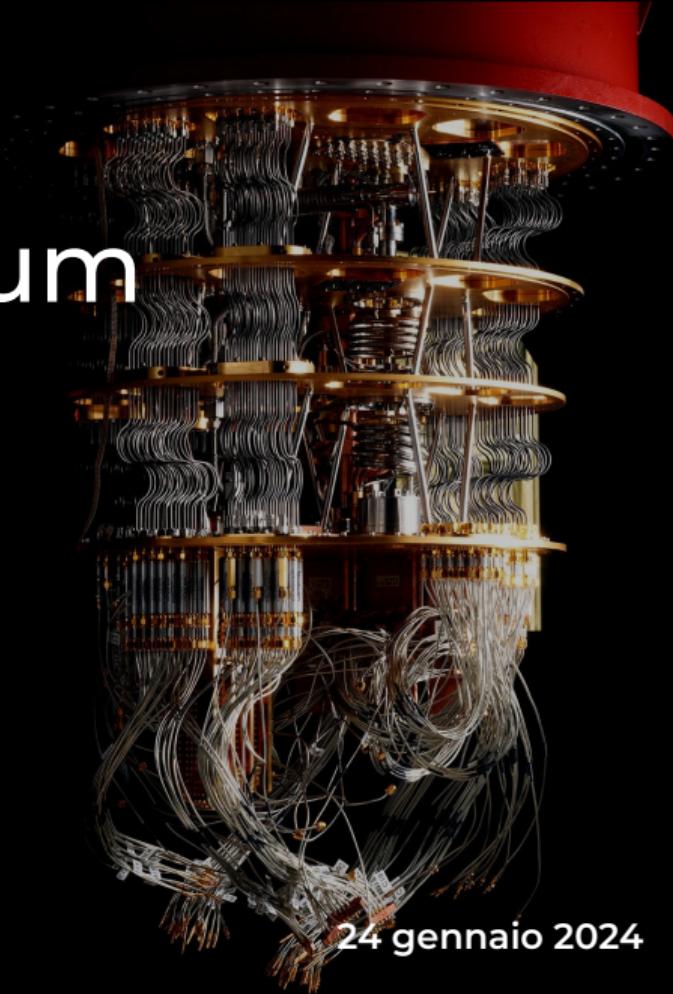


Esame di Deep Learning

# Variational Quantum Regressor

Simulazione e confronto con modelli classici

Edoardo Tronconi  
Matricola: 974734



24 gennaio 2024

# Obiettivi

# Perché quantum machine learning?

L'utilizzo di hardware quantistico per sviluppare architetture di machine learning ha diversi vantaggi:

- Basso consumo di energia per ottenere le previsioni
- La natura quantistica dei qubit permette una rappresentazione più profonda di alcuni sistemi rispetto a modelli classici
- Capacità di costruire modelli con molti meno parametri della controparte classica

# Obiettivi del Progetto

- Ottimizzare un VQR su diversi dataset
- Studiare le capacità di un VQR di rappresentare dataset di diversa complessità
- Confrontare il numero di parametri necessari per un VQR e un network neurale classico

# Contenuti

## 1 Pipeline di lavoro

- Preprocessing dei dati
- Ottimizzazione FFNN
- Ottimizzazione VQR
- Analisi risultati

## 2 Risultati per diversi dataset

- Test
- Concentrazioni CO<sub>2</sub>
- Housing
- Cars

# Pipeline

# Data Preprocessing

Prima di essere passati ai regressori i dati sono sottoposti ad alcuni step di preprocessing:

- Selezionate le features più significative
- Standardizzazione per features numeriche
- One-hot encoding per features categoriche
- Rescaling output a range [0, 1]

# FFNN optimization

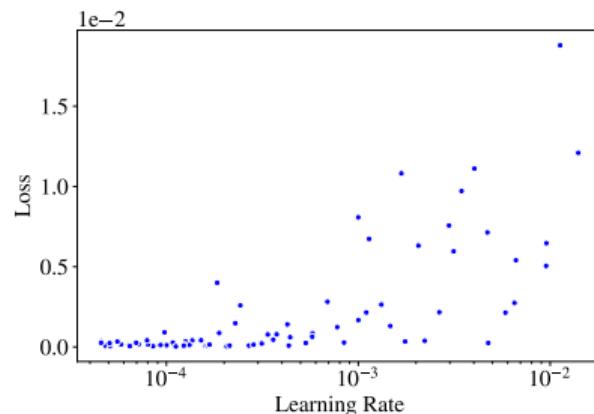
È stato allenato un network neurale feed forward ottimizzando con Hyperopt gli iperparametri di:

## 1 Architettura

- Numero di Layers → da 2 a 25
- Numero di Nodi per Layer → da 10 a  $10^4$

## 2 Ottimizzatore (Adam)

- Learning Rate → da  $5 \cdot 10^{-5}$  a  $10^{-1}$   
→ Dopo alcuni test fissato a  $10^{-4}$



# VQR optimization

È stato allenato un Variational Quantum Regressor ottimizzando:

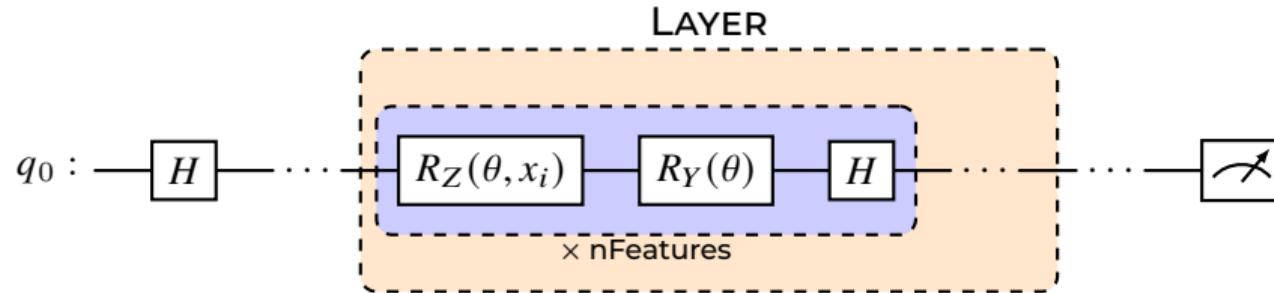
- Architettura dell'Ansatz
- Numero di Layers nell'Ansatz

# VQR optimization

## Ansatz Types

Sono state testate tre diverse architetture:

- 1 Ansatz Lineare ("linear" o "abs")



La differenza tra *linear* e *abs* sta nel modo in cui sono parametrizzati i gates:

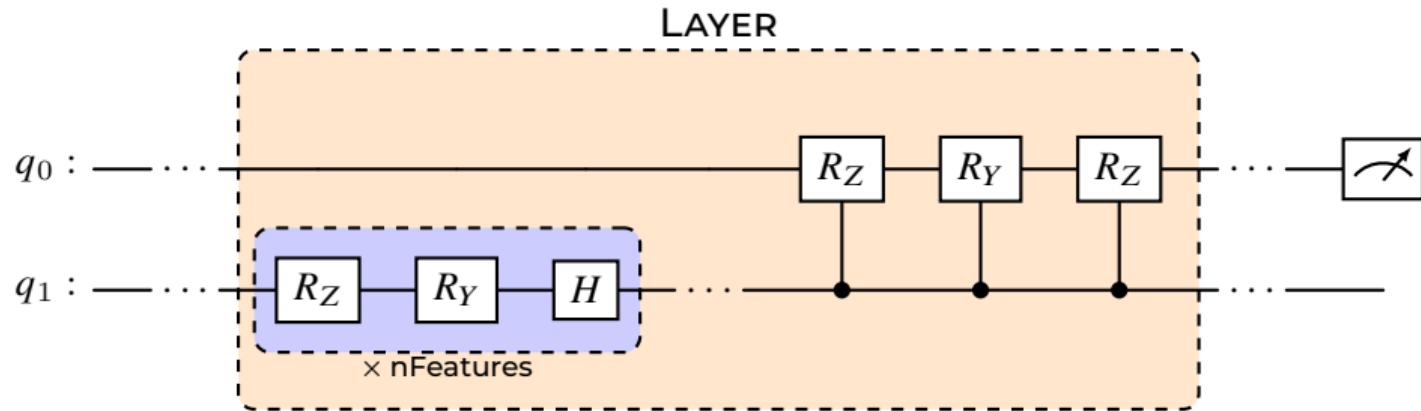
$$\textbf{Linear: } \theta_{R_Z} = a + b \cdot x_i, \quad \theta_{R_Y} = c$$

$$\textbf{Abs: } \theta_{R_Z} = a + b \cdot |x_i - c|, \quad \theta_{R_Y} = d$$

# VQR optimization

## Ansatz Types

### ② Ansatz a 2 qubit



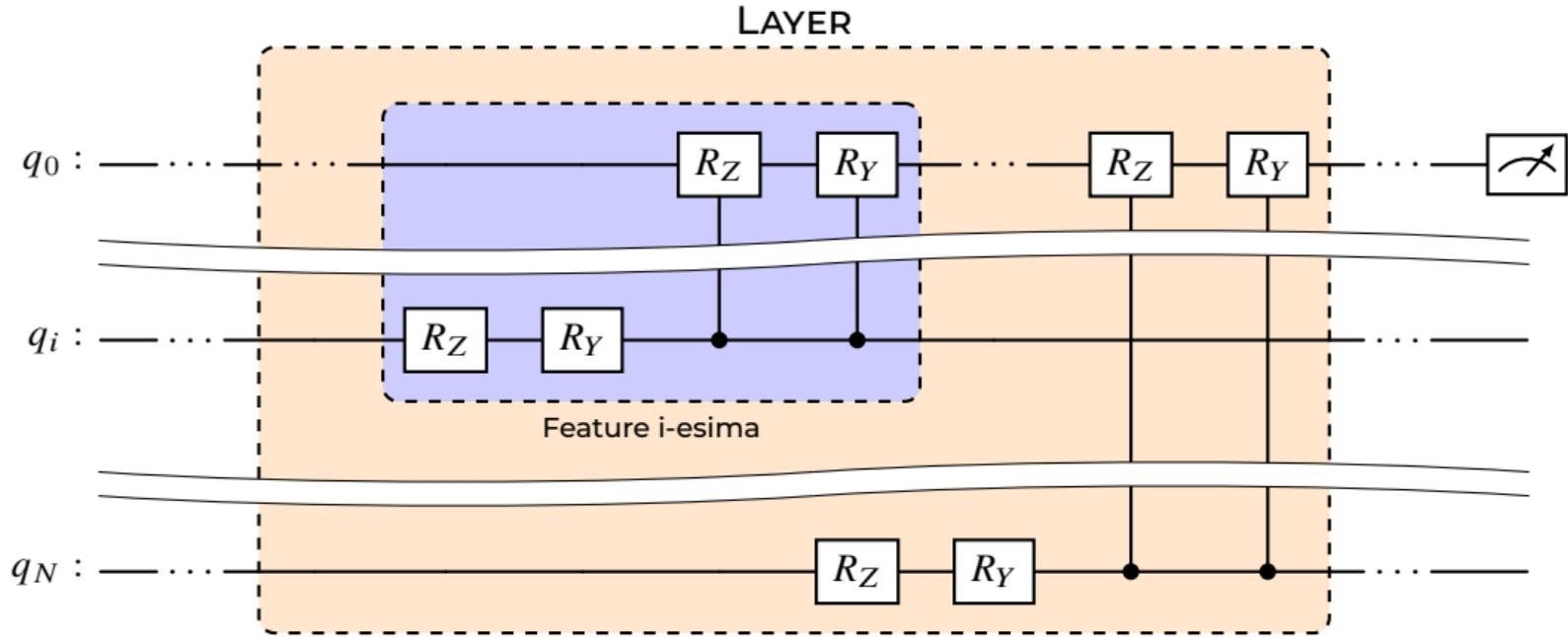
I gates in  $q_1$  sono parametrizzati come nel modello Lineare, quelli in  $q_0$  non dipendono dall'input  $x$

# VQR optimization

## Ansatz Types

### ③ Ansatz Deep

→ 1 qubit di output, un qubit di "controllo" per ogni feature



# Evaluation

La performance dei modelli è stata valutata sui dataset di test

- Calcolo di MSE e dell'errore medio assoluto
- Plot di previsioni al variare del valore vero
- Numero di parametri del modello ottimale

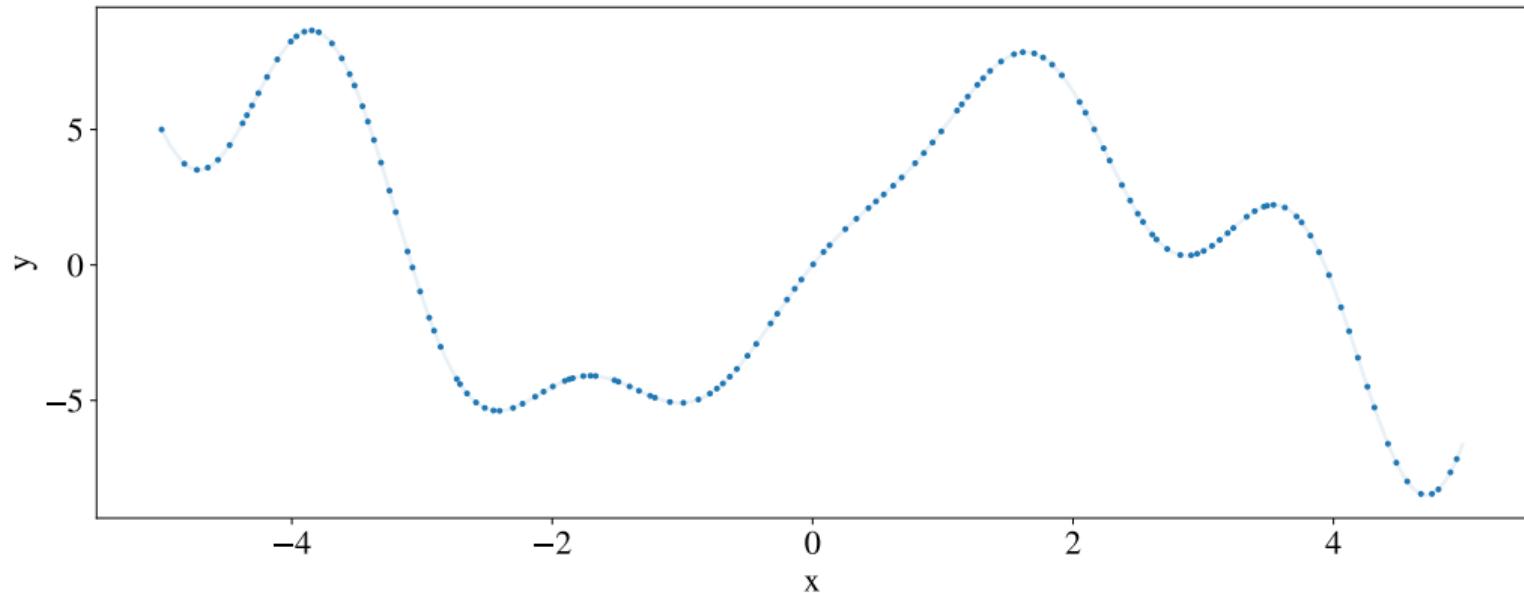
# Risultati

# **Funzione di Test**

# Funzione di test

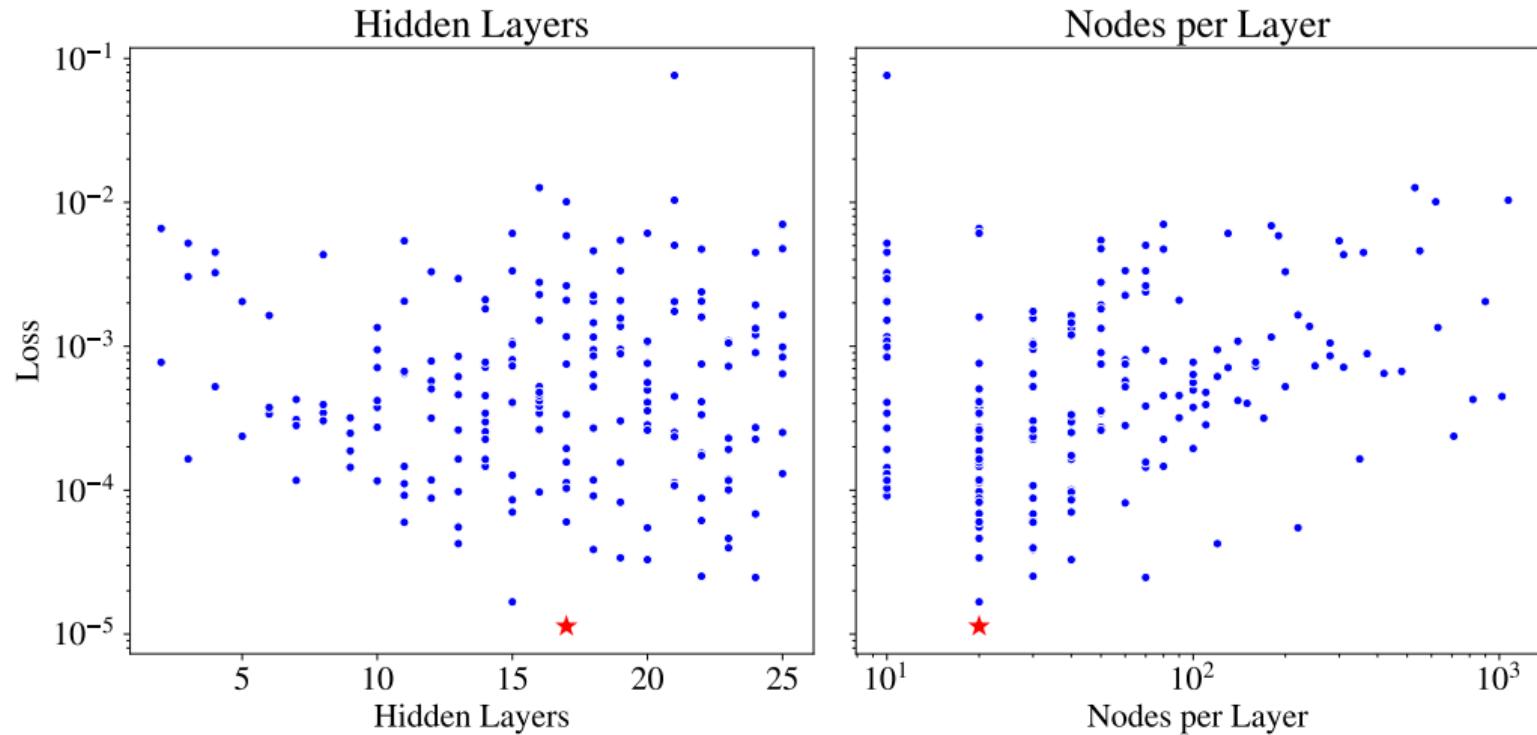
## Dataset

Per testare la pipeline sono stati campionati 1000 punti nell'intervallo  $[-5, 5]$  dalla funzione  $y = 6 \sin(x) + \frac{x^2}{10} - x \cdot \sin(3x)$



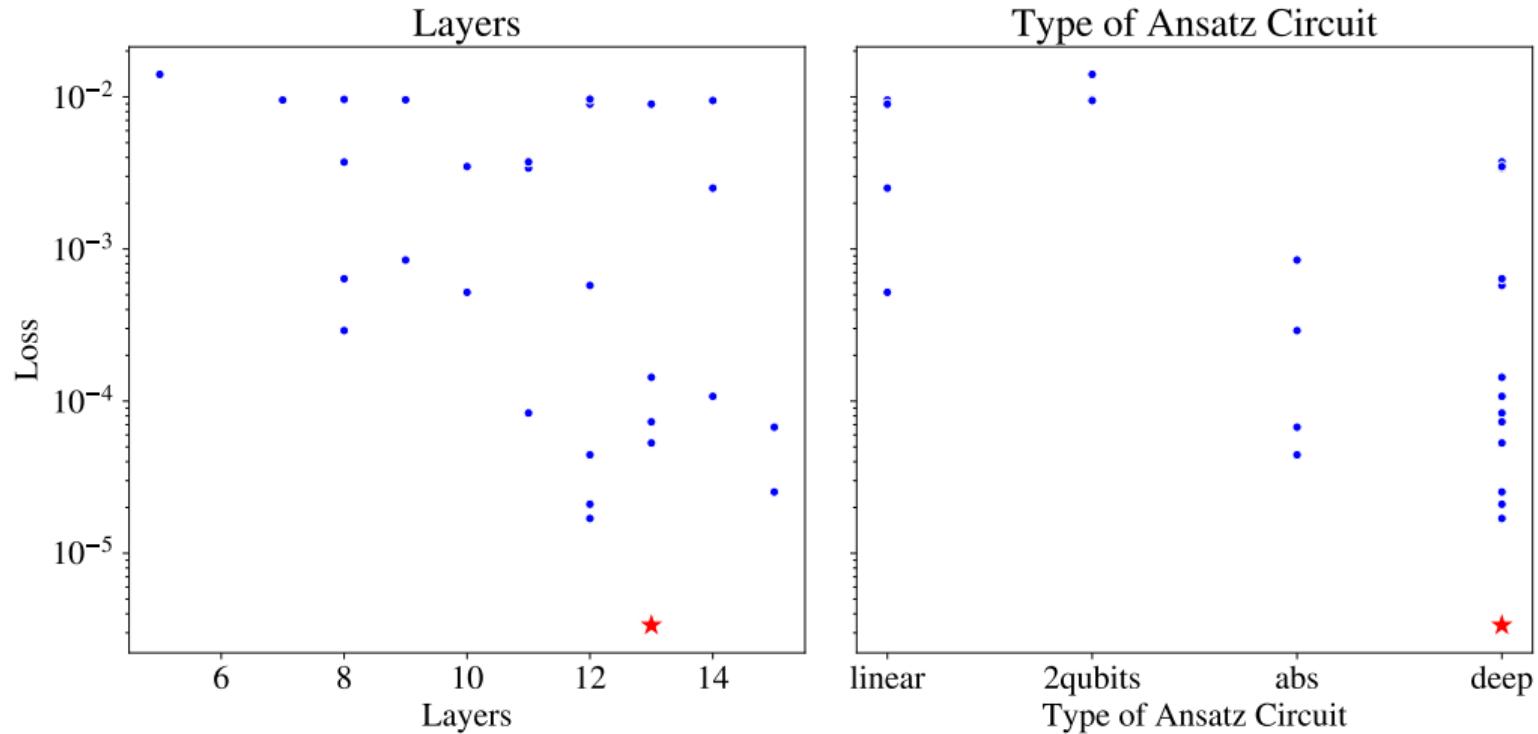
# Funzione di Test

## FFNN



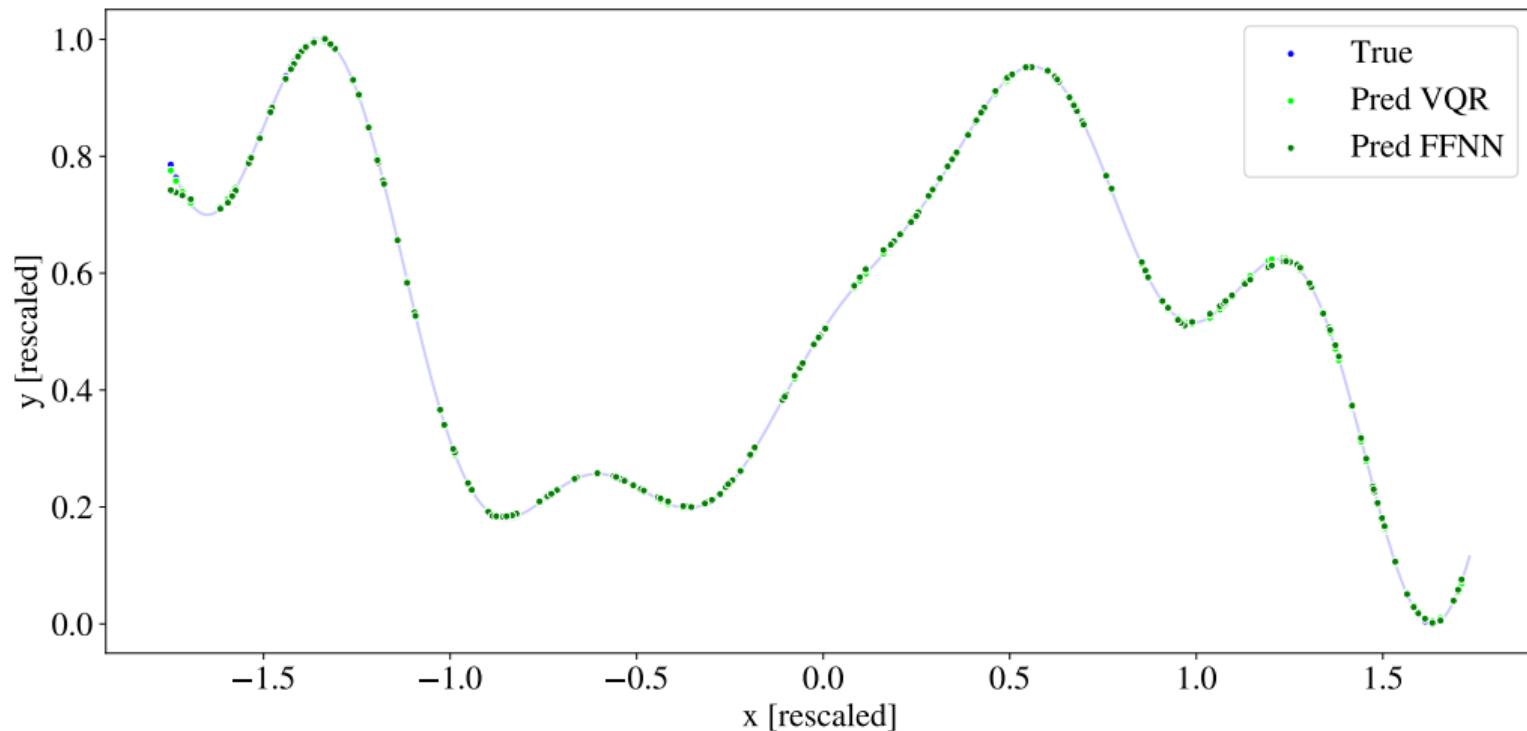
# Funzione di Test

## VQR



# Funzione di test

## Previsioni



# Funzione di test

## Risultati

- Entrambi i modelli hanno appreso la funzione di test, ma il modello quantistico ha molti meno parametri ed è riuscito a riprodurre la funzione più correttamente!

### Classico:

- Numero di parametri: 7681
- MSE Loss:  $2.21 \cdot 10^{-5}$
- Errore medio assoluto:  $2.59 \cdot 10^{-3}$

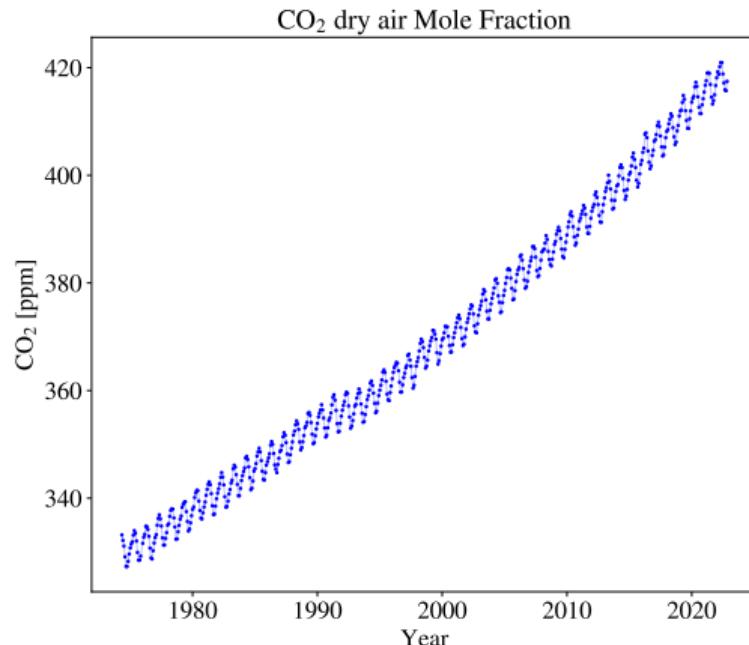
### Quantistico:

- Numero di parametri: 78
- MSE Loss:  $3.73 \cdot 10^{-6}$
- Errore medio assoluto:  $1.49 \cdot 10^{-3}$

# Concentrazioni di CO<sub>2</sub>

Dati del Mauna Loa Observatory relativi alle concentrazioni medie mensili di CO<sub>2</sub> dal 1974 al 2022. (<https://gml.noaa.gov/ccgg/trends/>)

- Valore medio in crescita di circa 2 ppm all'anno
  - Oscillazioni con periodo di un anno
- I modelli sono in grado di catturare sia la crescita annua che le variazioni mensili?



Periodo di misura descritto come:

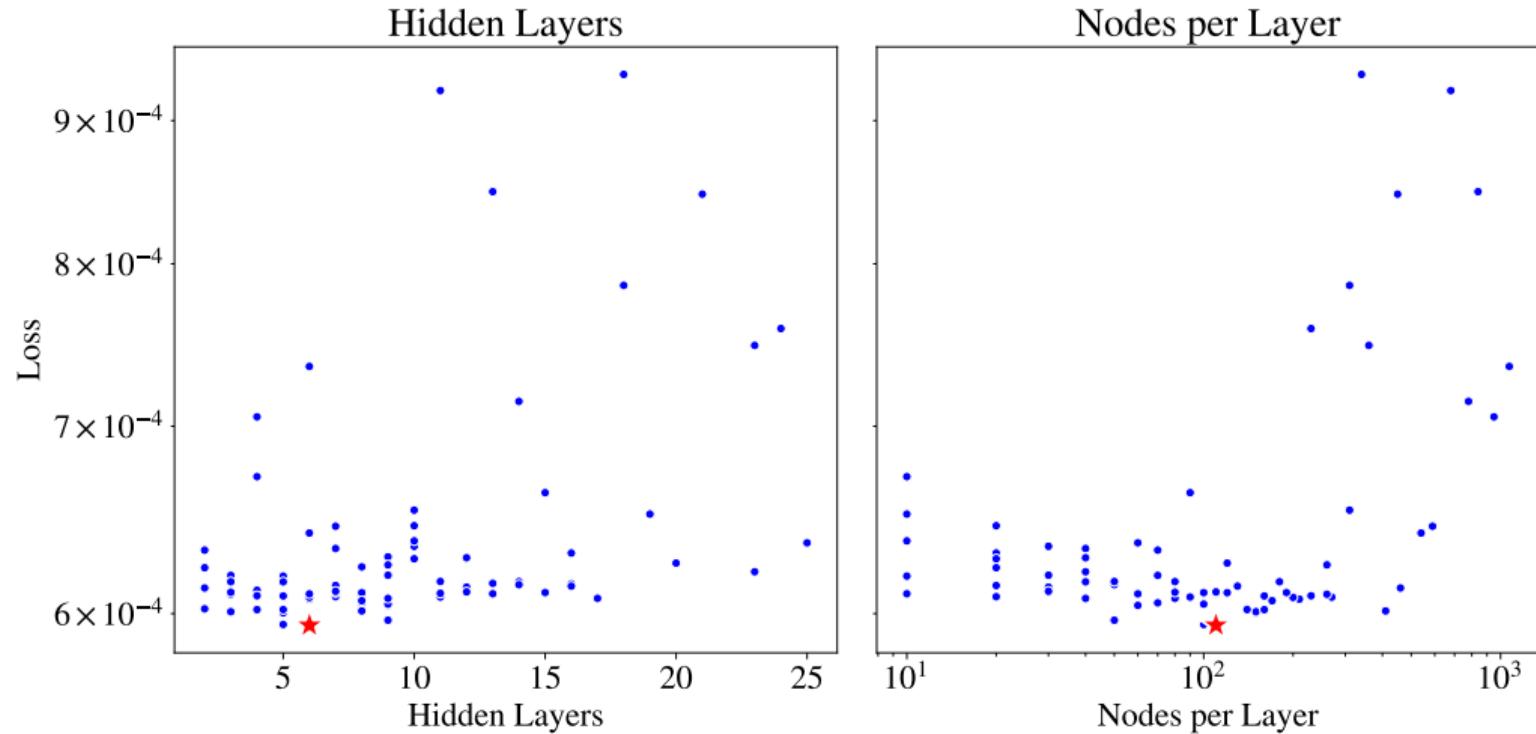
- Coppia "anno, mese"
- Anno decimale

→ Allenato algoritmo con ciascuna delle due possibilità come input

	Year	Month	Time Decimal	Value	Value STD
9	1974	5	1974.328767	333.16	0.36
10	1974	6	1974.413699	332.17	0.41
11	1974	7	1974.495890	331.11	0.49
12	1974	8	1974.580822	329.11	0.64
13	1974	9	1974.665753	327.30	0.64
...	...	...	...	...	...
587	2022	7	2022.495890	418.85	1.02
588	2022	8	2022.580822	417.18	0.68
589	2022	9	2022.665753	415.89	0.58
590	2022	10	2022.747945	415.72	0.48
591	2022	11	2022.832877	417.47	0.82

582 rows × 5 columns

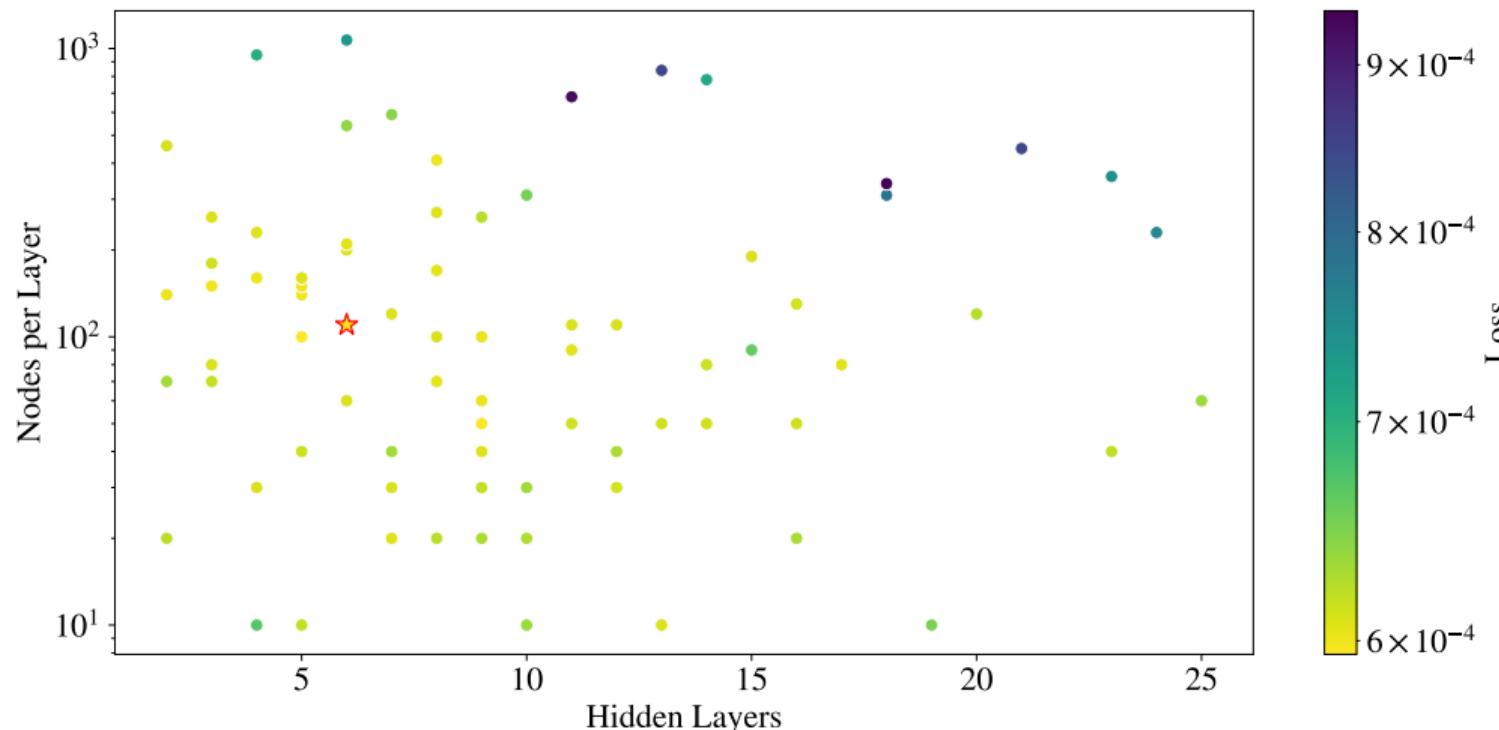
# $\text{CO}_2$ - 1 variable FFNN



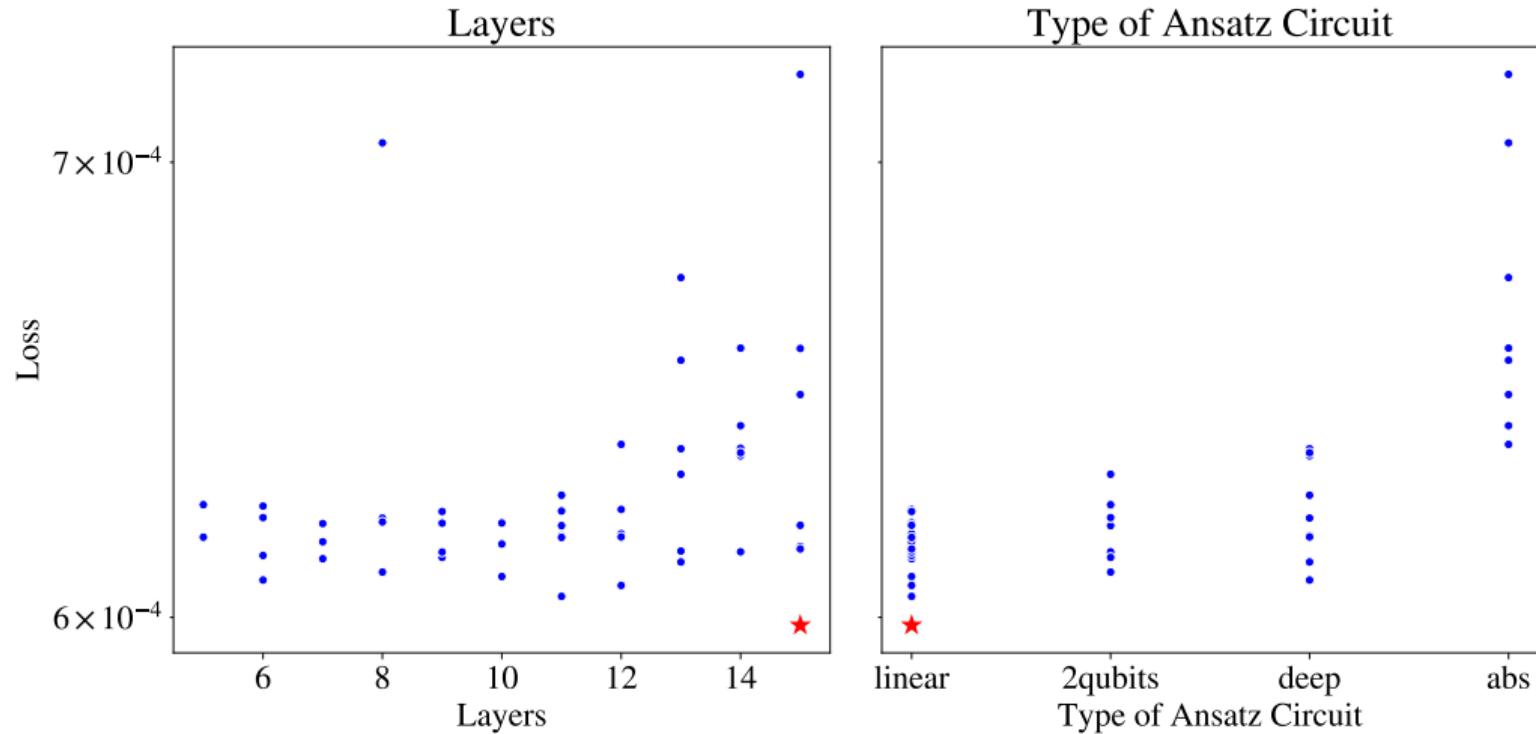
# CO<sub>2</sub> - 1 variable

## FFNN

Poca dipendenza dall'architettura, il modello non riesce a migliorare oltre un certo livello

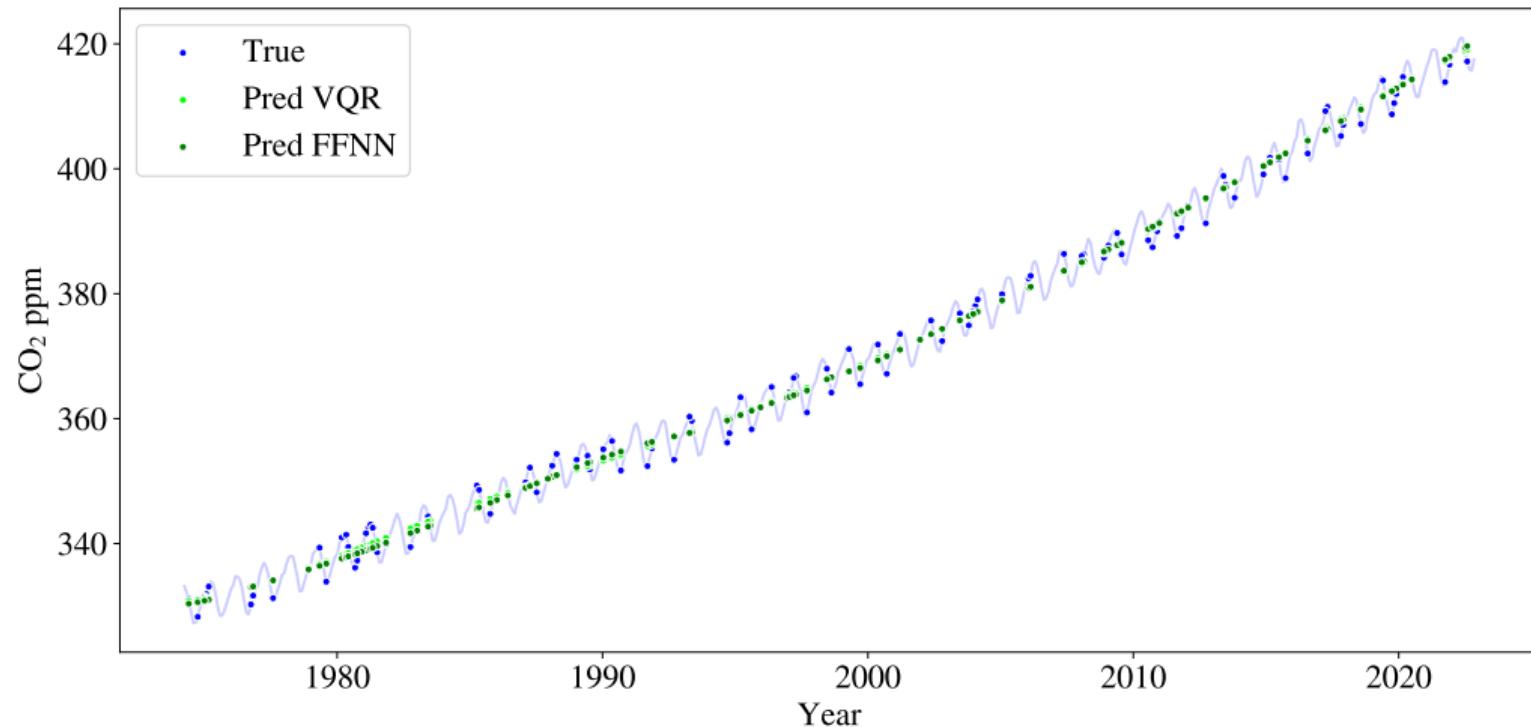


# $\text{CO}_2$ - 1 variable VQR

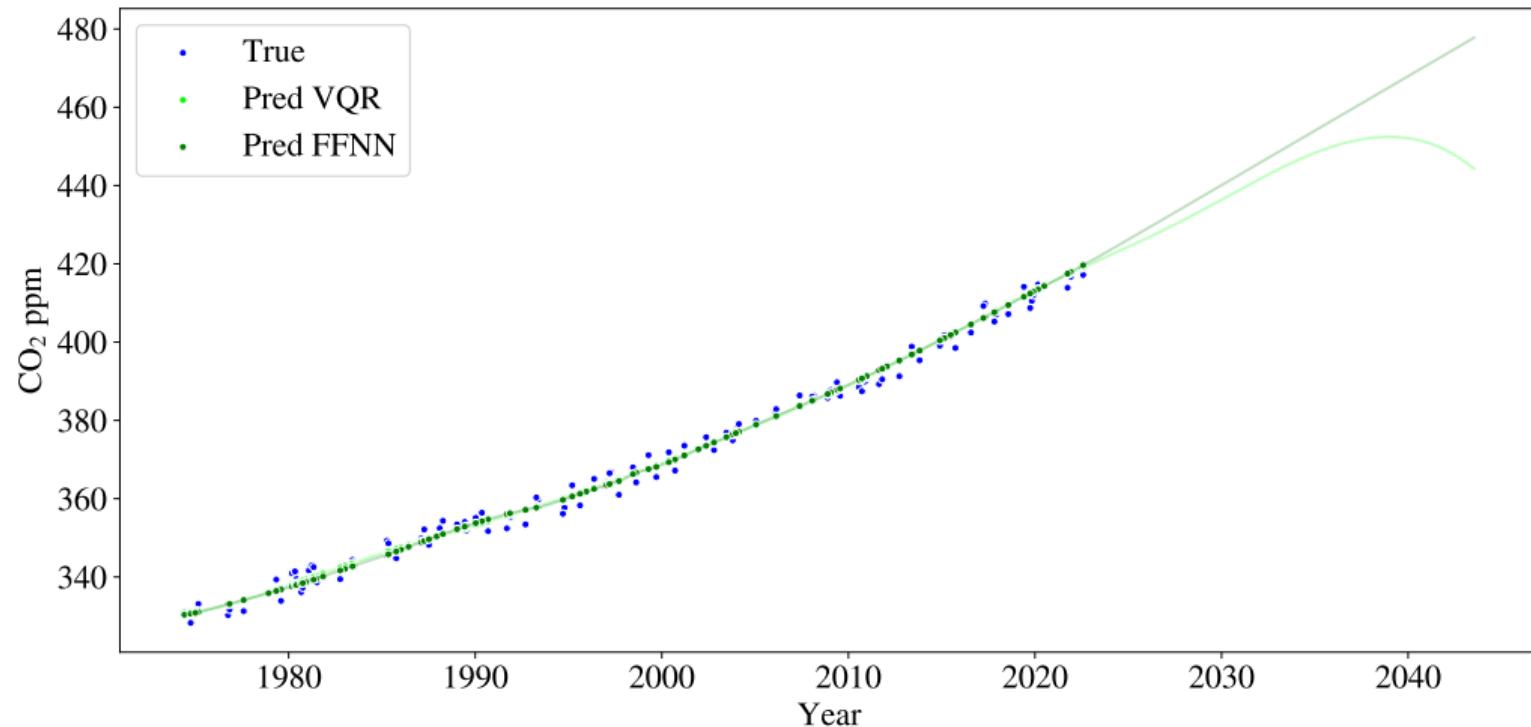


# CO<sub>2</sub> - 1 variable

## Previsioni



# CO<sub>2</sub> - 1 variable Estrapolazione



- Entrambi i modelli catturano il trend in crescita, ma non sono in grado di riprodurre le oscillazioni mensili utilizzando solo l'anno decimale come input
- Sebbene non sia il modello più opportuno, il FFNN riesce ad estrapolare in modo ragionevole ad anni futuri.  
Il VQR è basato su rotazioni → non riesce ad estrarre, output oscillante

### Classico:

- Numero di parametri: 61381
- Loss:  $5.55 \cdot 10^{-4}$
- Errore medio assoluto:  $2.03 \cdot 10^{-2}$

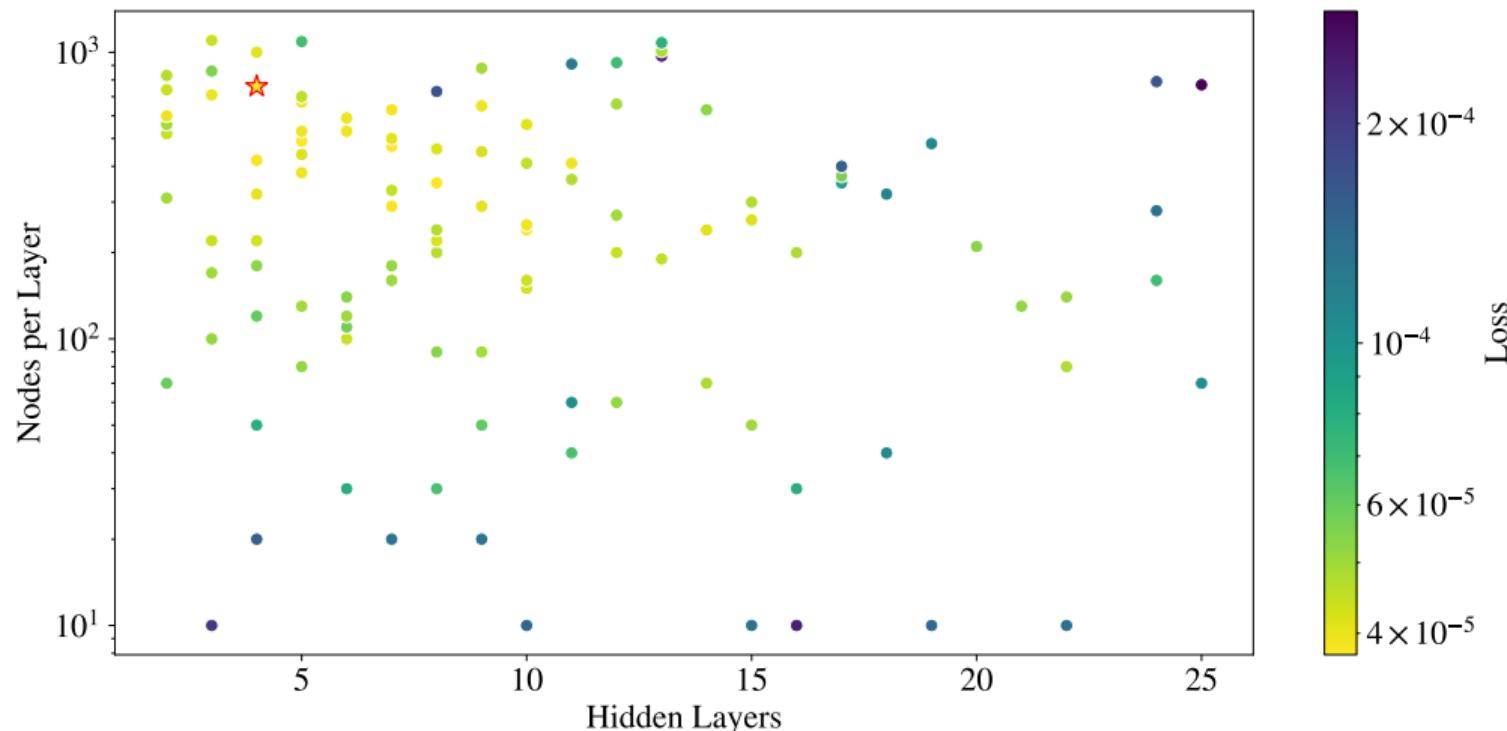
### Quantistico:

- Numero di parametri: 45
- Loss:  $5.30 \cdot 10^{-4}$
- Errore medio assoluto:  $1.99 \cdot 10^{-2}$

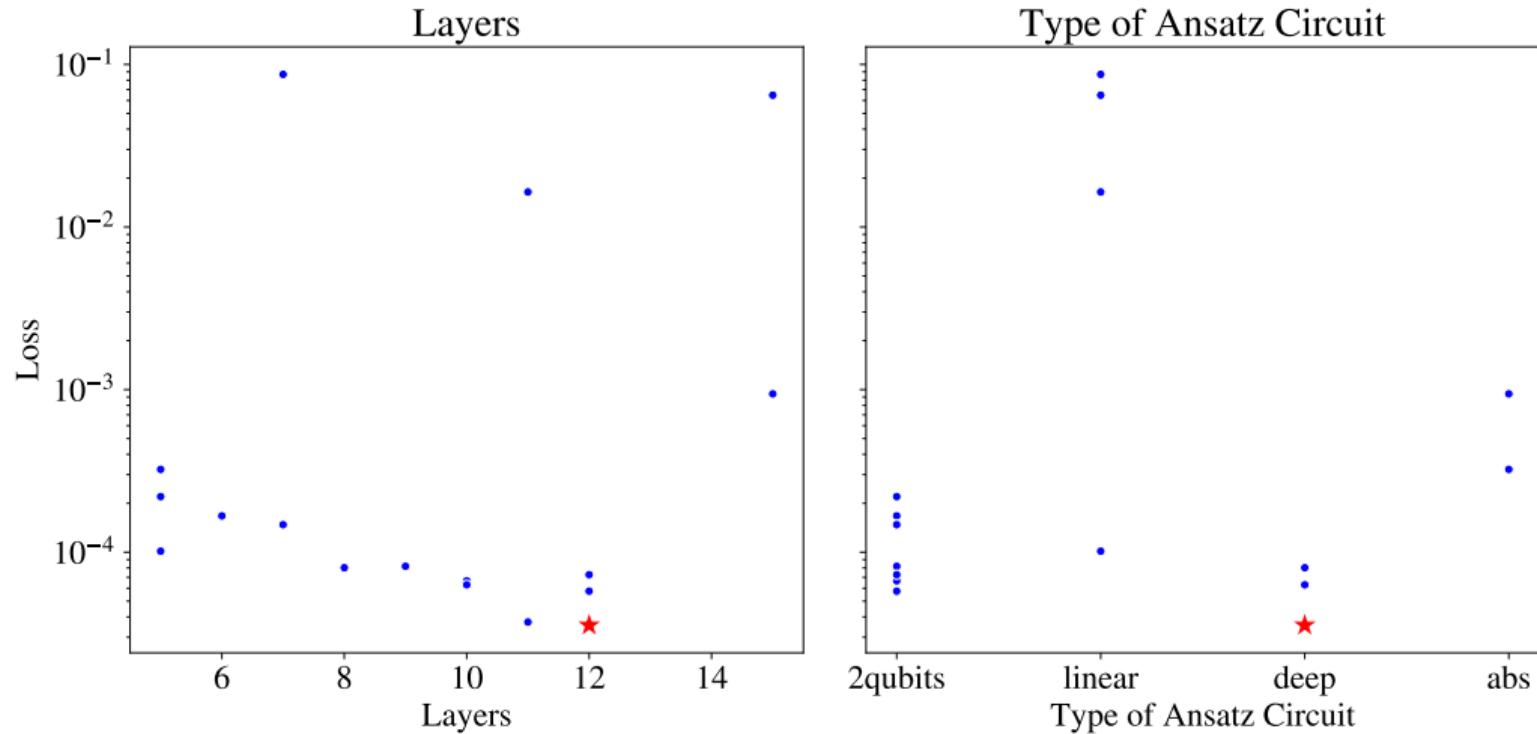
# CO<sub>2</sub> - 2 variabili FFNN

Dipendenza dall'architettura molto più marcata!

→ Si nota come Hyperopt concentra i punti nelle regioni più promettenti

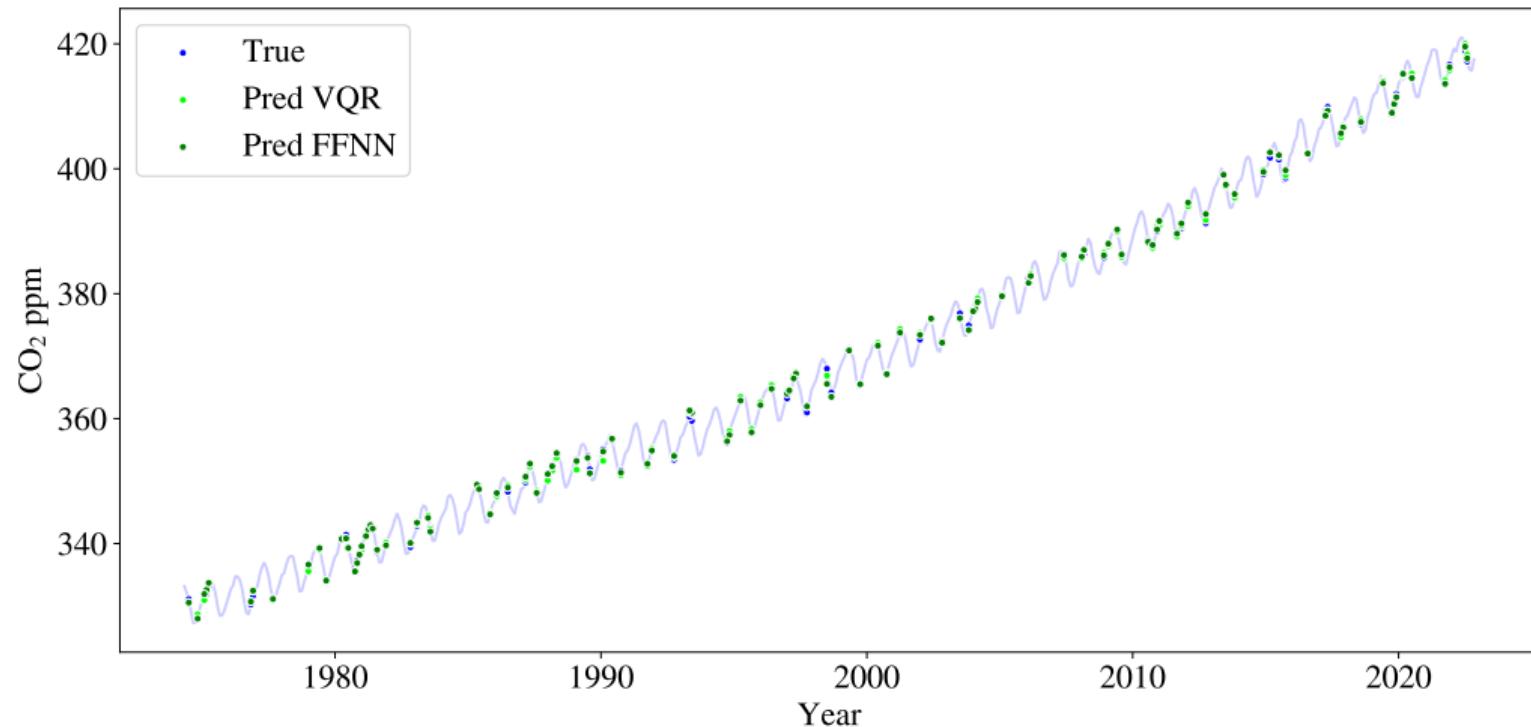


# $\text{CO}_2$ - 2 variabili VQR



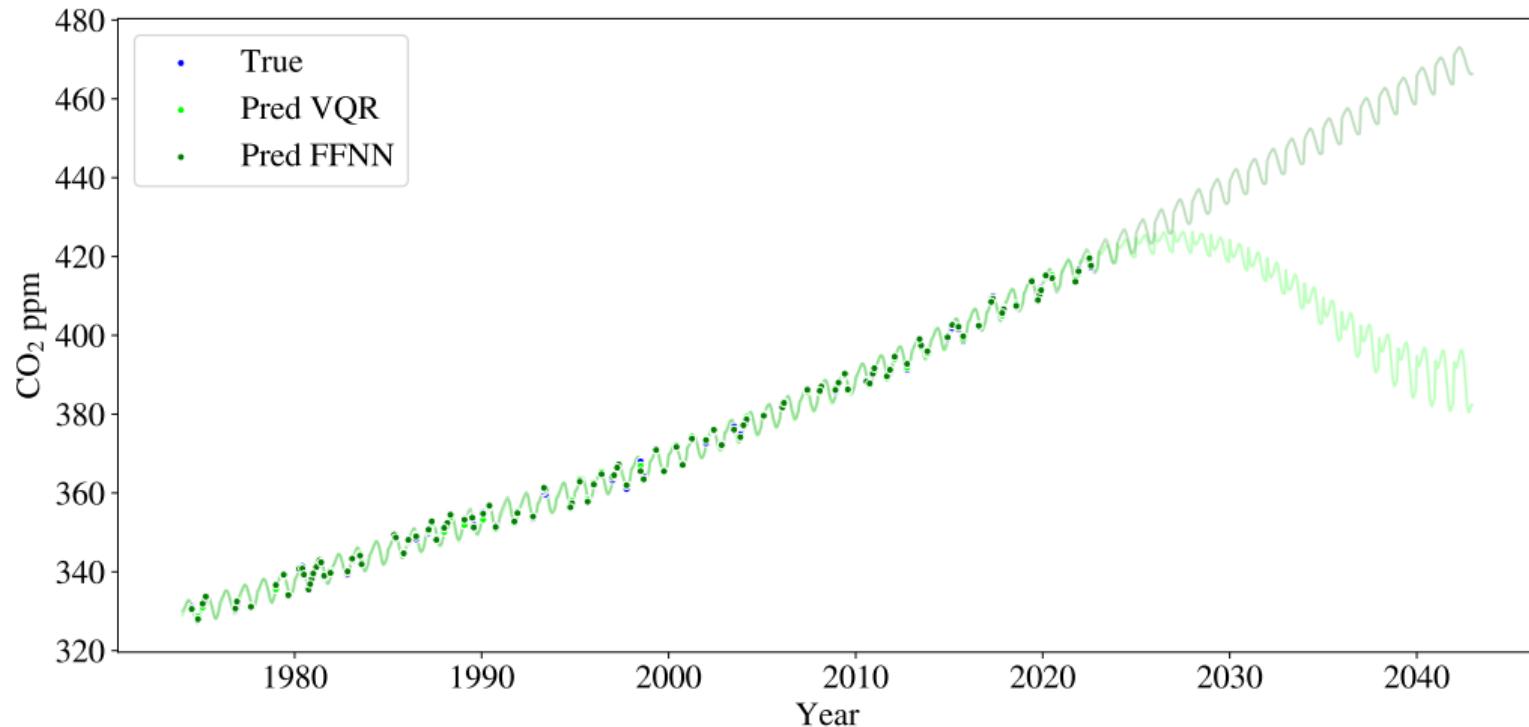
# CO<sub>2</sub> - 2 variabili

## Previsioni



# CO<sub>2</sub> - 2 variabili

## Estrapolazione



- Con due variabili di input sono riprodotti sia il trend in crescita che le oscillazioni
- Anche in questo caso si nota come il VQR abbia un comportamento oscillante se l'input è fuori dal range dei dati di training

### Classico:

- Numero di parametri: 61381
- Loss:  $5.55 \cdot 10^{-4}$
- Errore medio assoluto:  $4.61 \cdot 10^{-3}$

### Quantistico:

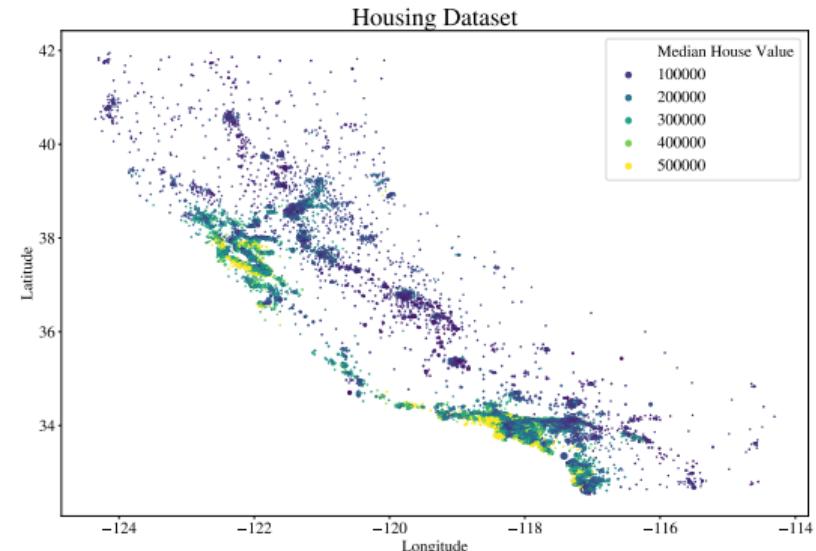
- Numero di parametri: 45
- Loss:  $5.30 \cdot 10^{-4}$
- Errore medio assoluto:  $4.97 \cdot 10^{-3}$

# Housing

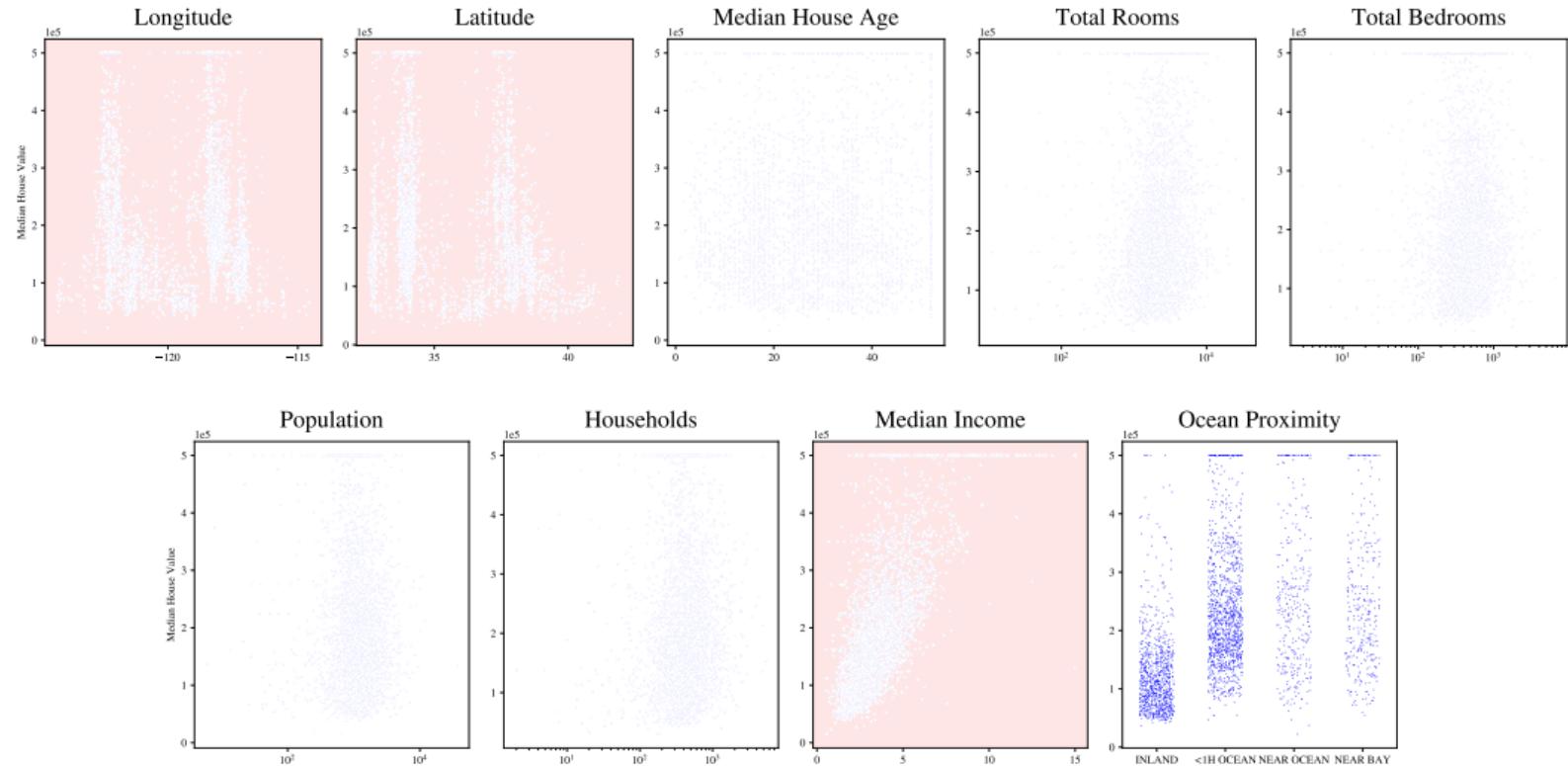
# Housing Dataset

Dataset di Kaggle contenente i valori mediani delle case nei distretti della California, ricavati dal censimento del 1990. (<https://www.kaggle.com/datasets/camnugent/california-housing-prices>)

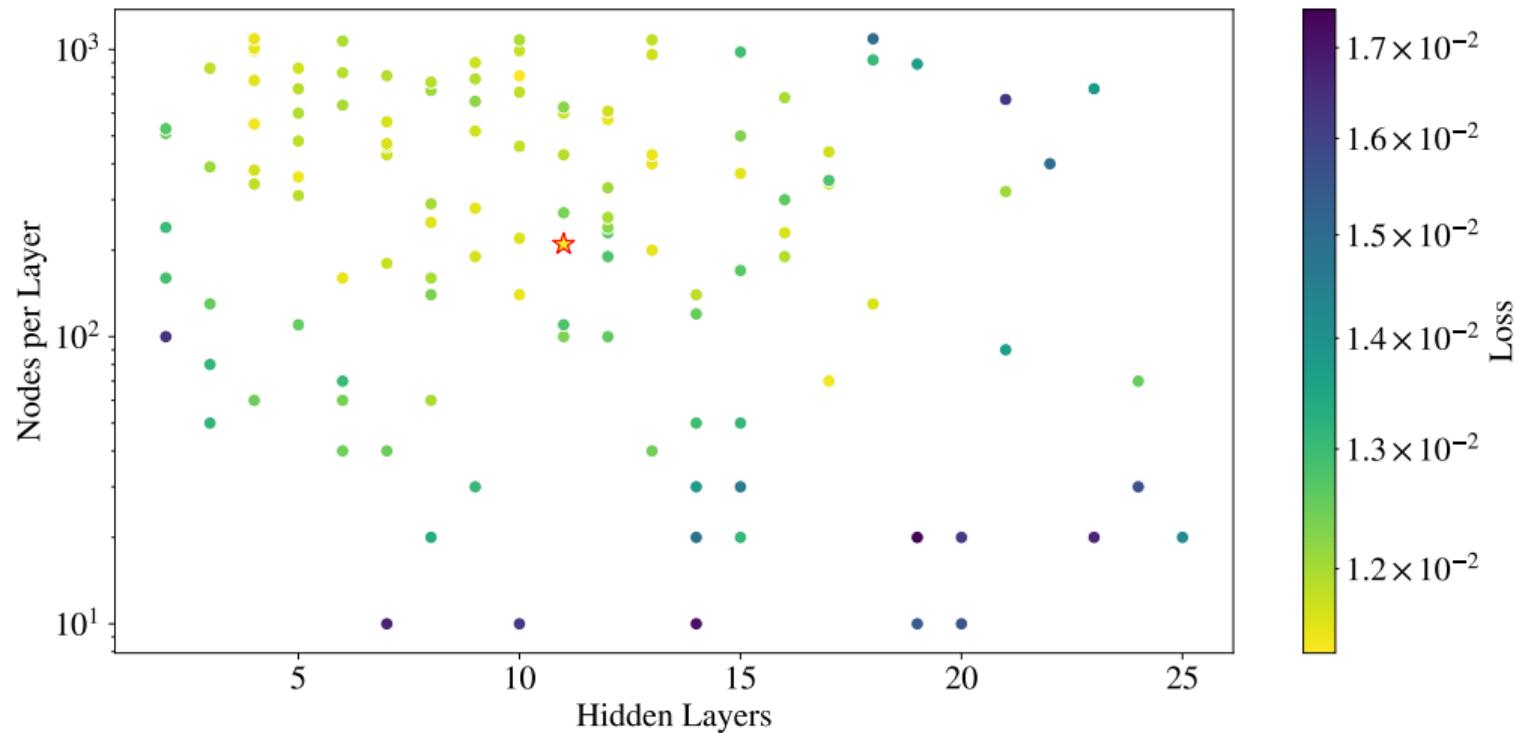
Circa 20000 entries, ciascuna delle quali ha 10 features (incluso il prezzo).



# Housing Feature Selection



# Housing FFNN



## Training VQR estremamente lento

Il dataset contiene moltissimi dati, è più complesso dei precedenti e ha più feature!

- Ciascuna previsione necessita di aggiornare i parametri ed eseguire il circuito → molti dati → molte esecuzioni
- Ciascuna iterazione dell'algoritmo di ottimizzazione calcola più volte la loss modificando un parametro per volta per stimare il gradiente
  - Poichè il dataset è più complesso sono necessarie architetture con molti layer → più parametri → molte esecuzioni

## Training VQR estremamente lento

Il dataset contiene moltissimi dati, è più complesso dei precedenti e ha più feature!

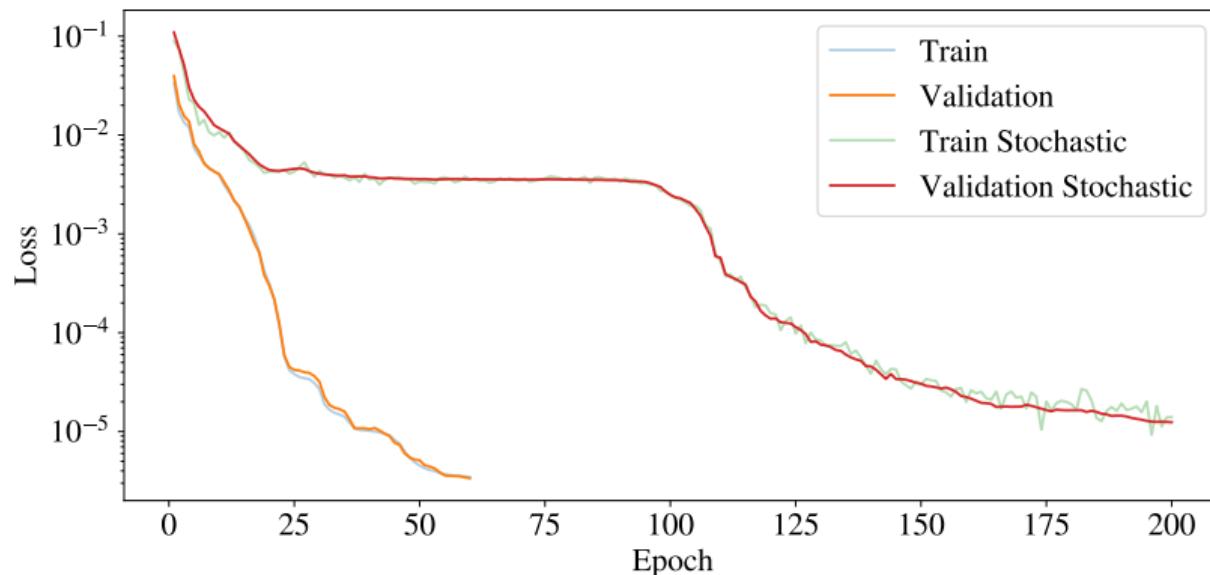
- Ciascuna previsione necessita di aggiornare i parametri ed eseguire il circuito → molti dati → molte esecuzioni
- Ciascuna iterazione dell'algoritmo di ottimizzazione calcola più volte la loss modificando un parametro per volta per stimare il gradiente
  - Poichè il dataset è più complesso sono necessarie architetture con molti layer → più parametri → molte esecuzioni

→ **STOCHASTIC GRADIENT DESCENT**

# VQR - Stochastic Gradient Descent

## Wrapper funzione training

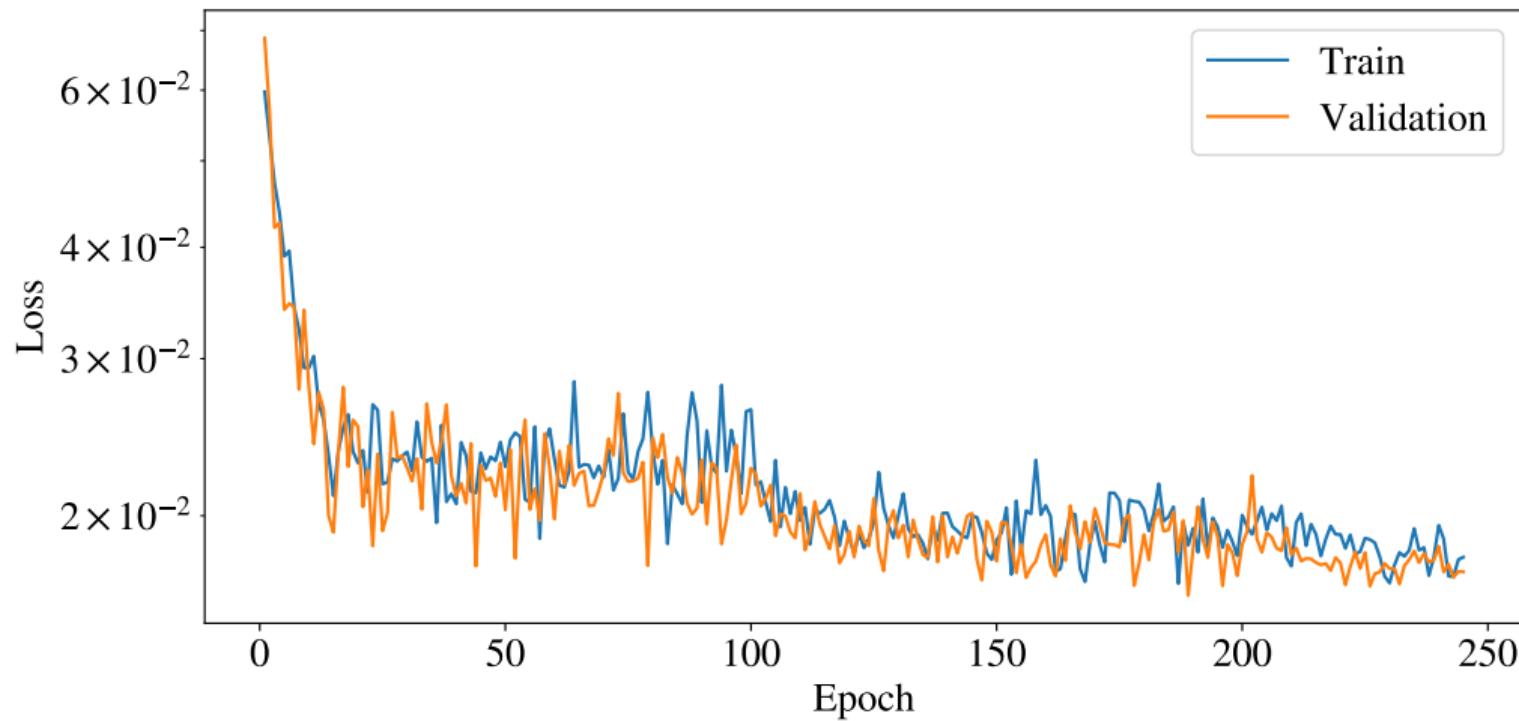
Ad ogni iterazione di `scipy.optimize.minimize` ricampionato un sottoinsieme casuale di dimensione fissata dal dataset di training



# Housing

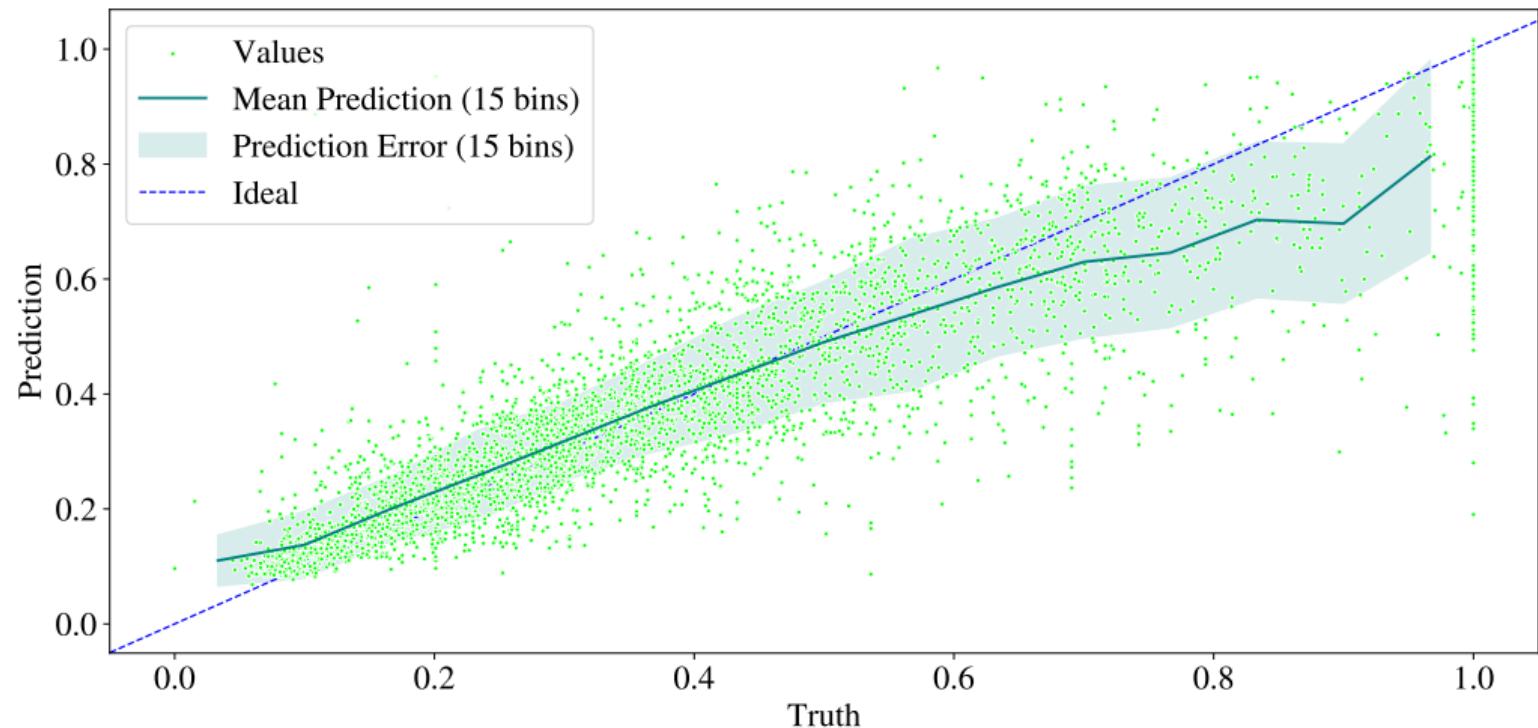
## VQR Loss History

Eseguito un solo training con architettura “deep” e 15 layers



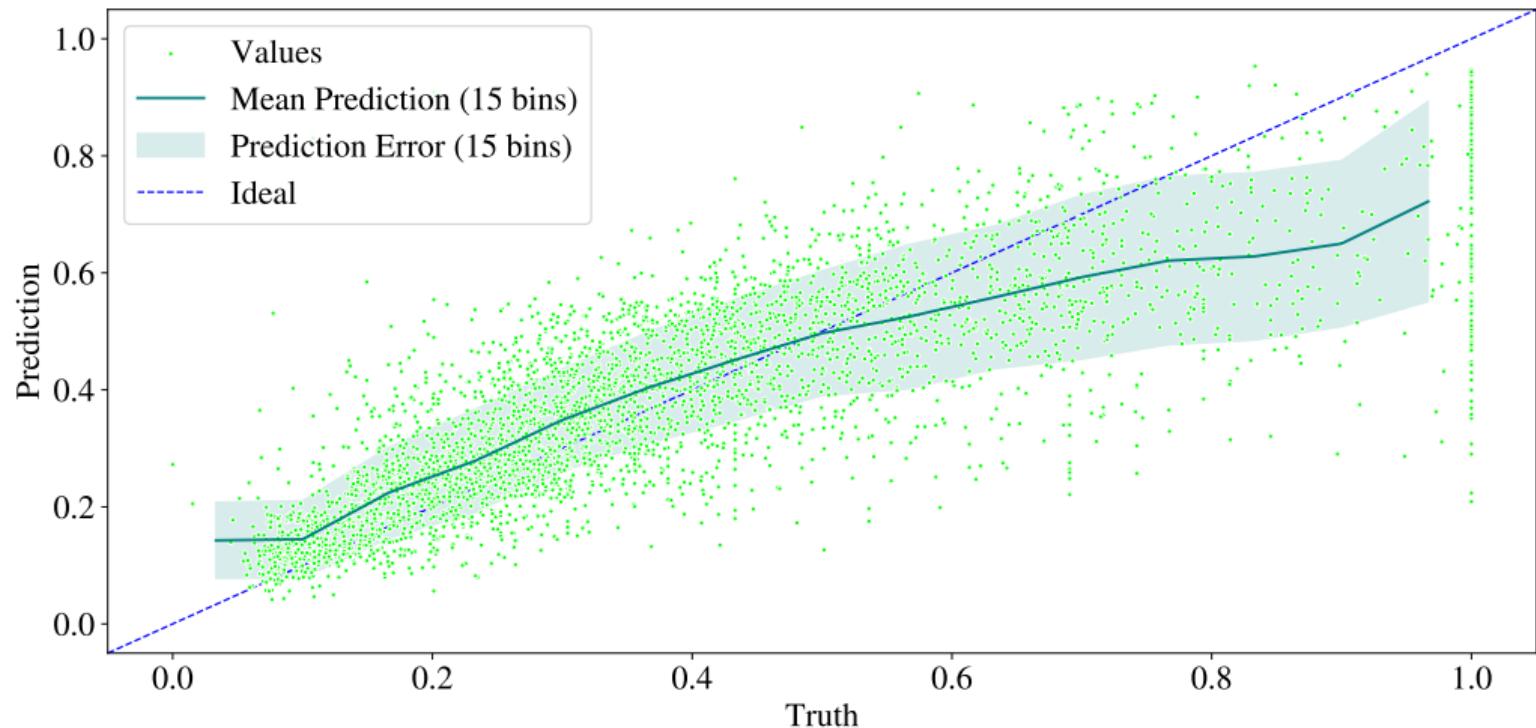
# Housing

## Previsioni FFNN



# Housing

## Previsioni VQR



Anche nel caso di un dataset più complesso il modello quantistico ha una performance confrontabile con quello classico, e molti meno parametri

### Classico:

- Numero di parametri: 444151
- MSE Loss: 0.0129
- Errore medio assoluto: 0.0756

### Quantistico:

- Numero di parametri: 270
- MSE Loss: 0.0192
- Errore medio assoluto: 0.0986

# Conclusioni

# Conclusioni

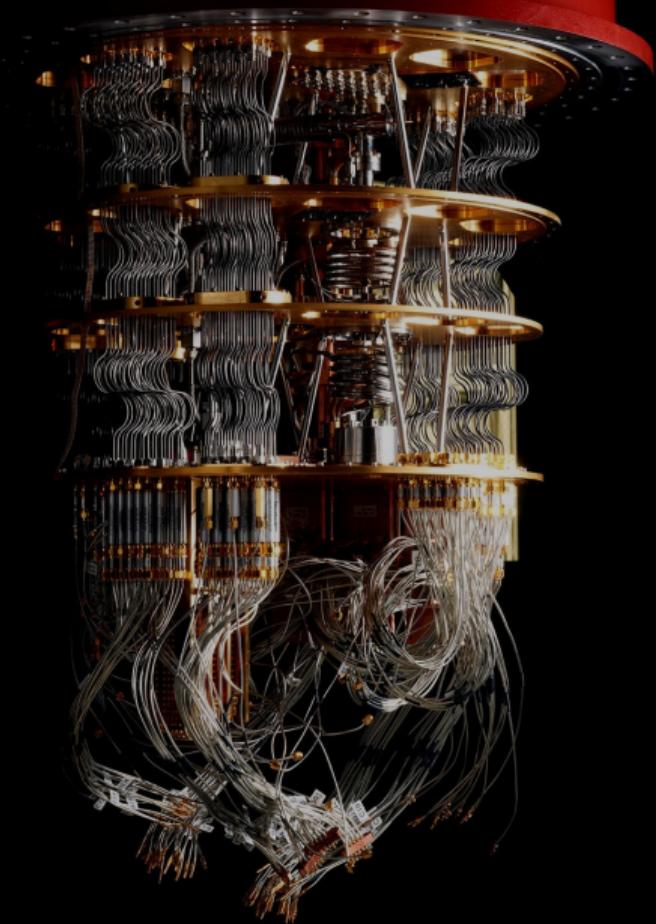
- ① VQR è in grado di riprodurre diversi dataset in modo confrontabile o migliore rispetto a un FFNN, utilizzando un numero estremamente inferiore di parametri
- ② Per modelli complessi i tempi di allenamento diventano molto più elevati  
→ ottimizzazione difficile

## NUMERO DI PARAMETRI:

DATASET	Test	CO <sub>2</sub> (1var)	CO <sub>2</sub> (2var)	Housing	Cars
FFNN	6781	61381	964321	444151	4361
VQR	78	45	144	270	540
VARIAZIONE	-98.8%	-99.9%	-99.99%	-99.93%	-87.6%

## LOSS:

DATASET	Test	CO <sub>2</sub> (1var)	CO <sub>2</sub> (2var)	Housing	Cars
FFNN	0.000021	0.000555	0.000036	0.013125	0.007436
VQR	0.000004	0.000530	0.000039	0.019229	0.013276
VARIAZIONE	-82.3%	-4.4%	+7.6%	+48.52%	+78.5%



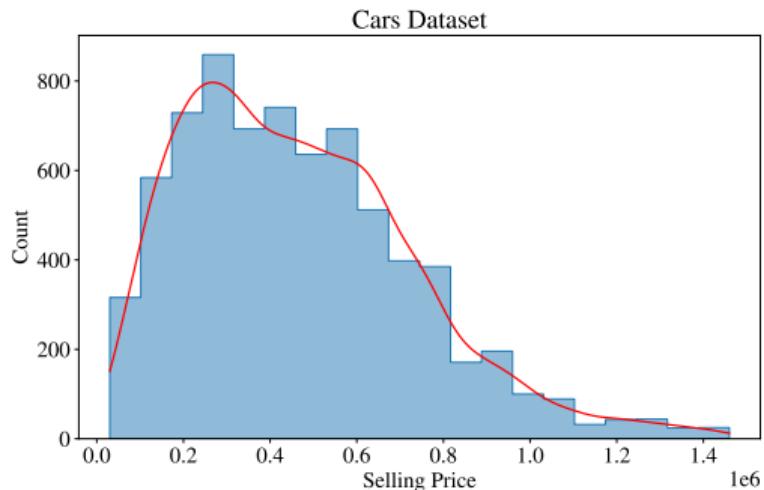
# Backup

# Dataset Cars

# Cars

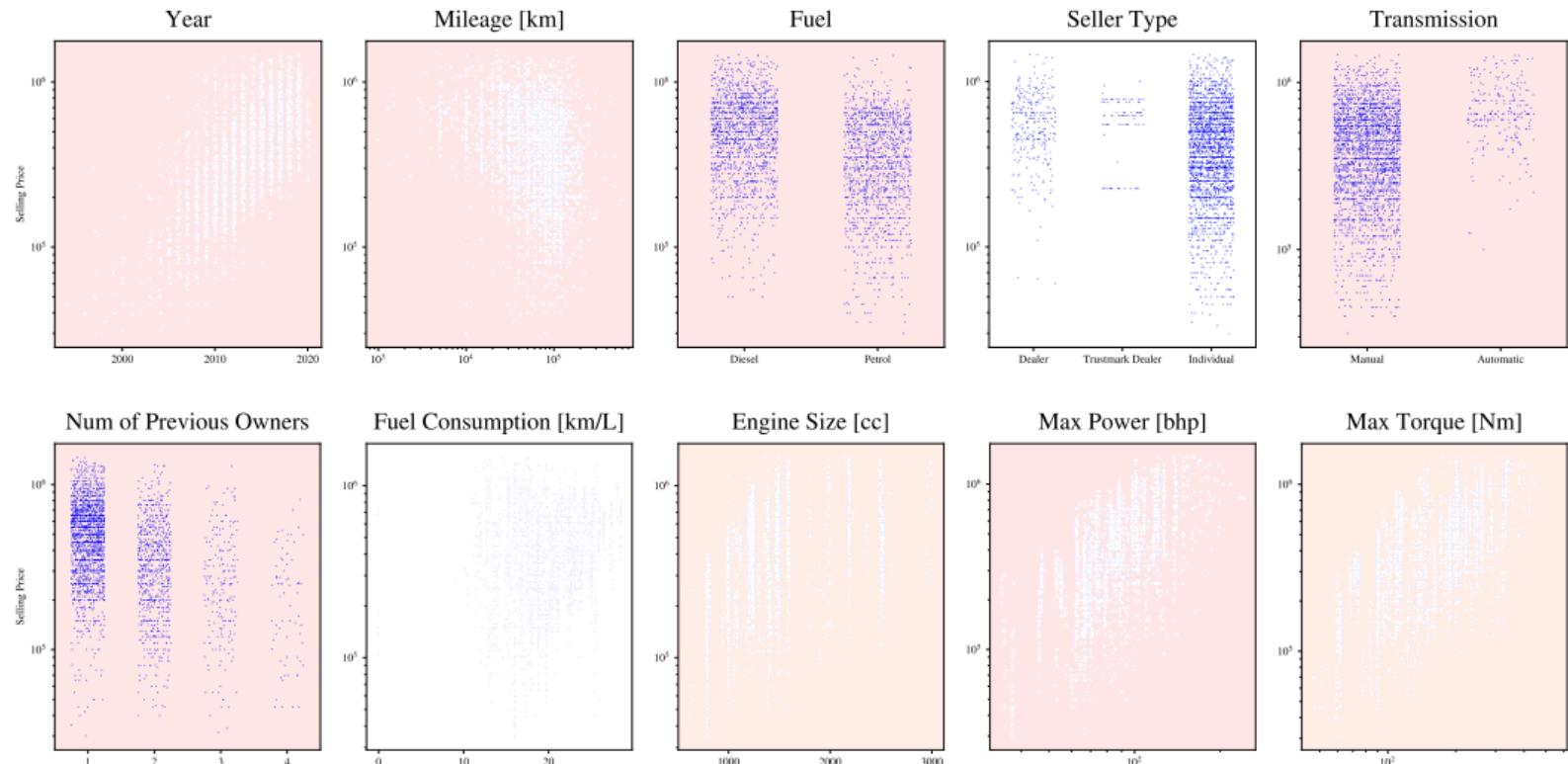
Dataset di Kaggle contenente il prezzo di vendita di auto usate.  
(<https://www.kaggle.com/datasets/nehalbirla/vehicle-dataset-from-cardekho>)

Circa 7000 entries, ciascuna delle quali ha 12 features (incluso il prezzo).



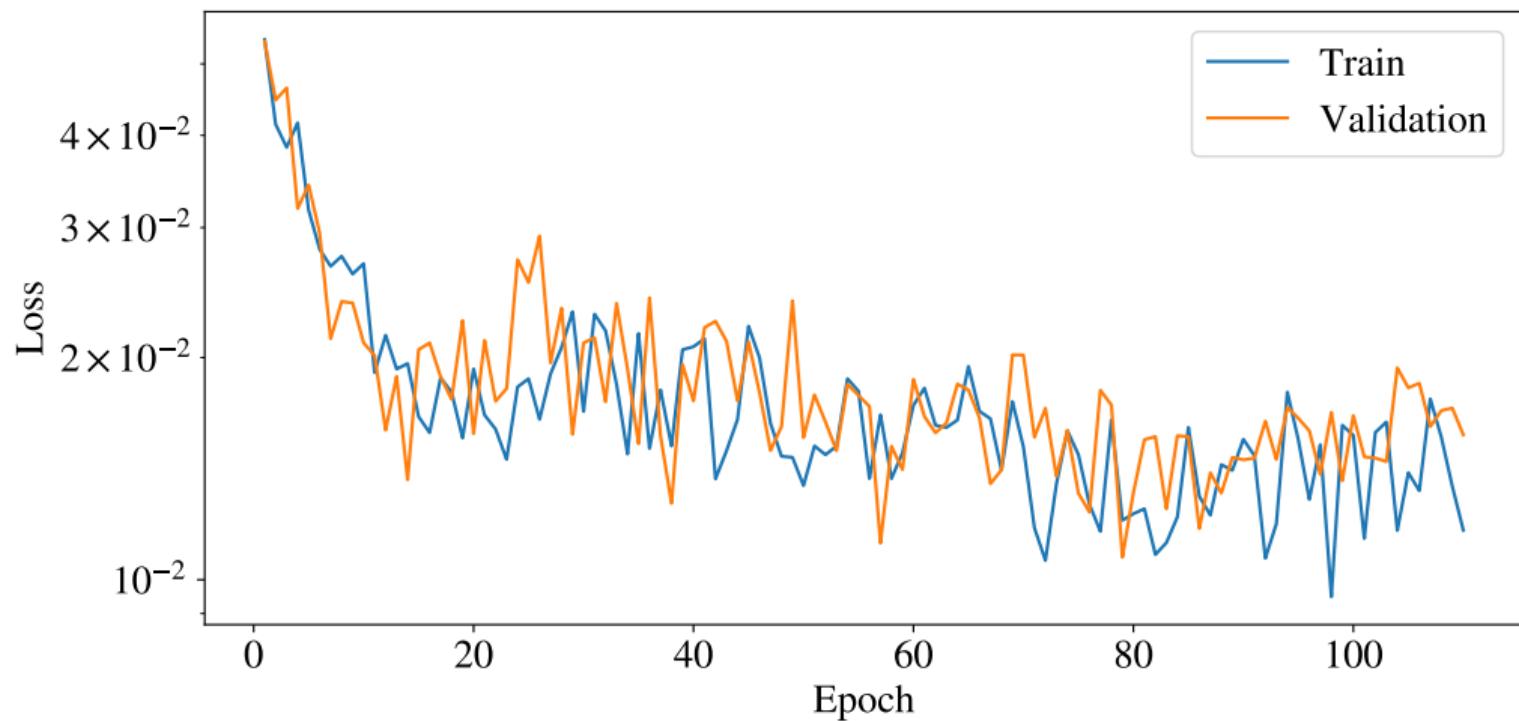
# Cars

## Feature Selection



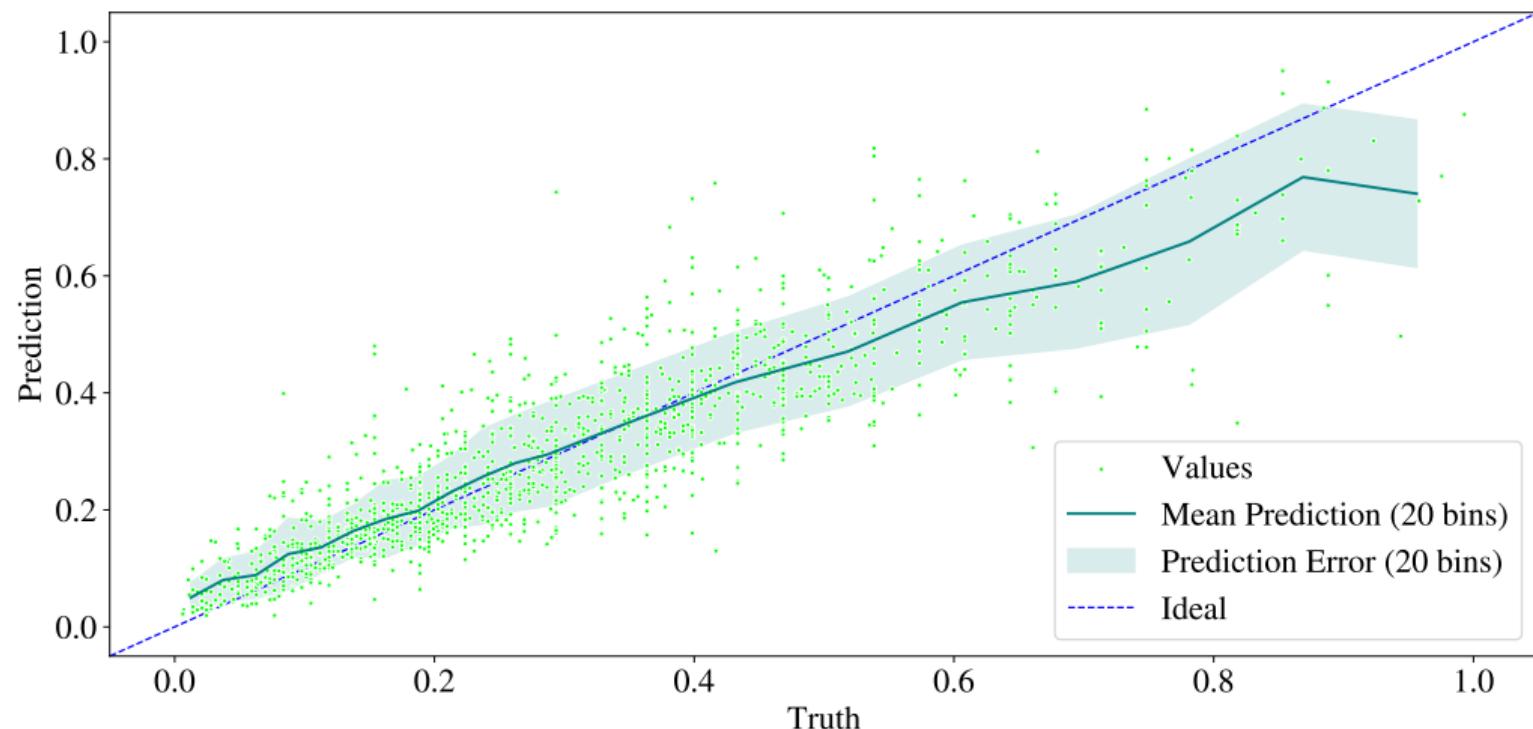
# Cars

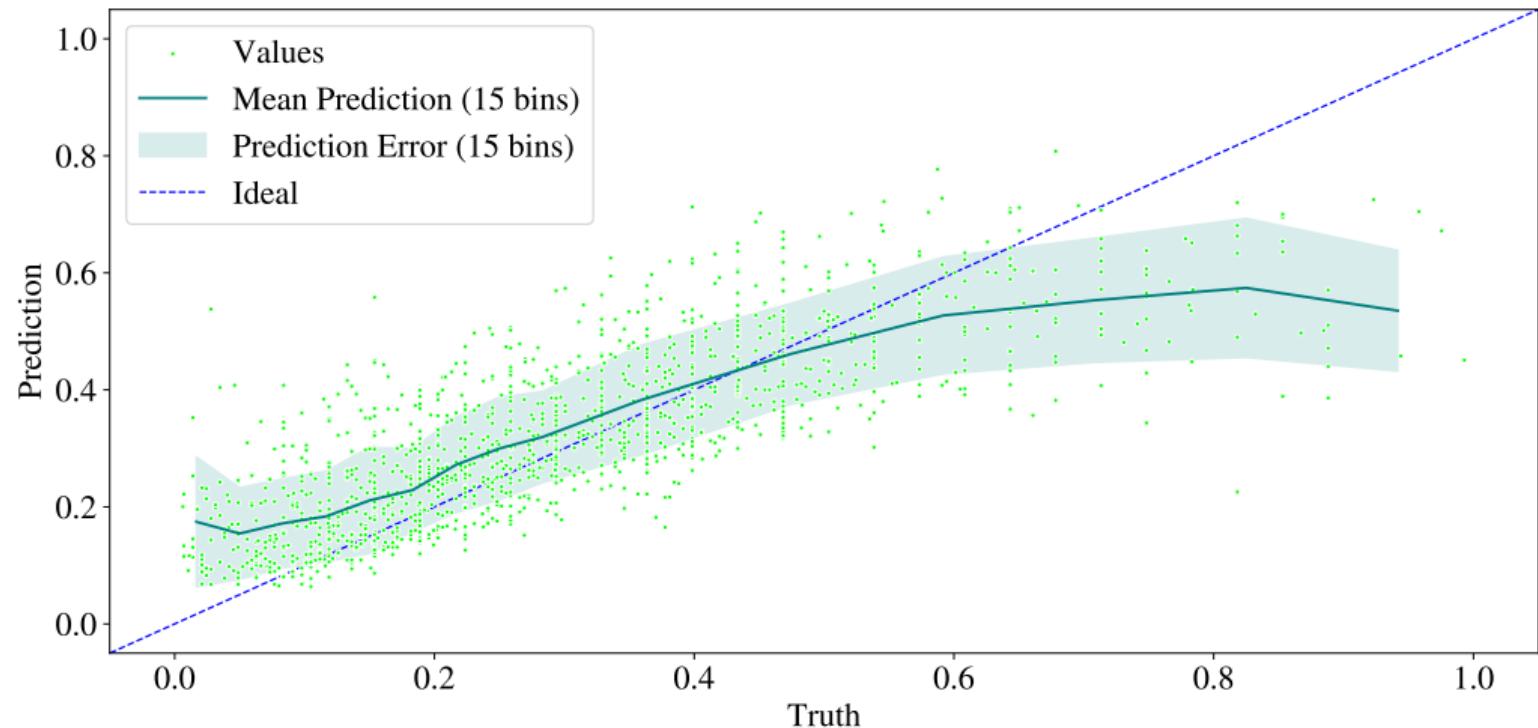
## VQR Loss History



# Cars

## Previsioni FFNN





In questo caso il modello quantistico non riesce ad ottenere ottimi risultati.

La performance potrebbe migliorare con:

- Più epochhe di training
- Una diversa ansatz

### Classico:

- Numero di parametri: 4361
- MSE Loss: 0.0074
- Errore medio assoluto: 0.0612

### Quantistico:

- Numero di parametri: 540
- MSE Loss: 0.0133
- Errore medio assoluto: 0.0848

# Selezione Features

# Feature Selection

## Housing

	Median House Value	Longitude	Latitude	Median House Age	Total Rooms	Total Bedrooms	Population	Households	Median Income	Ocean Proximity
0	452600.0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	NEAR BAY
1	358500.0	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	NEAR BAY
2	352100.0	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	NEAR BAY
3	341300.0	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	5.6431	NEAR BAY
4	342200.0	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	3.8462	NEAR BAY
...	...	...	...	...	...	...	...	...	...	...
20635	78100.0	-121.09	39.48	25.0	1665.0	374.0	845.0	330.0	1.5603	INLAND
20636	77100.0	-121.21	39.49	18.0	697.0	150.0	356.0	114.0	2.5568	INLAND
20637	92300.0	-121.22	39.43	17.0	2254.0	485.0	1007.0	433.0	1.7000	INLAND
20638	84700.0	-121.32	39.43	18.0	1860.0	409.0	741.0	349.0	1.8672	INLAND
20639	89400.0	-121.24	39.37	16.0	2785.0	616.0	1387.0	530.0	2.3886	INLAND

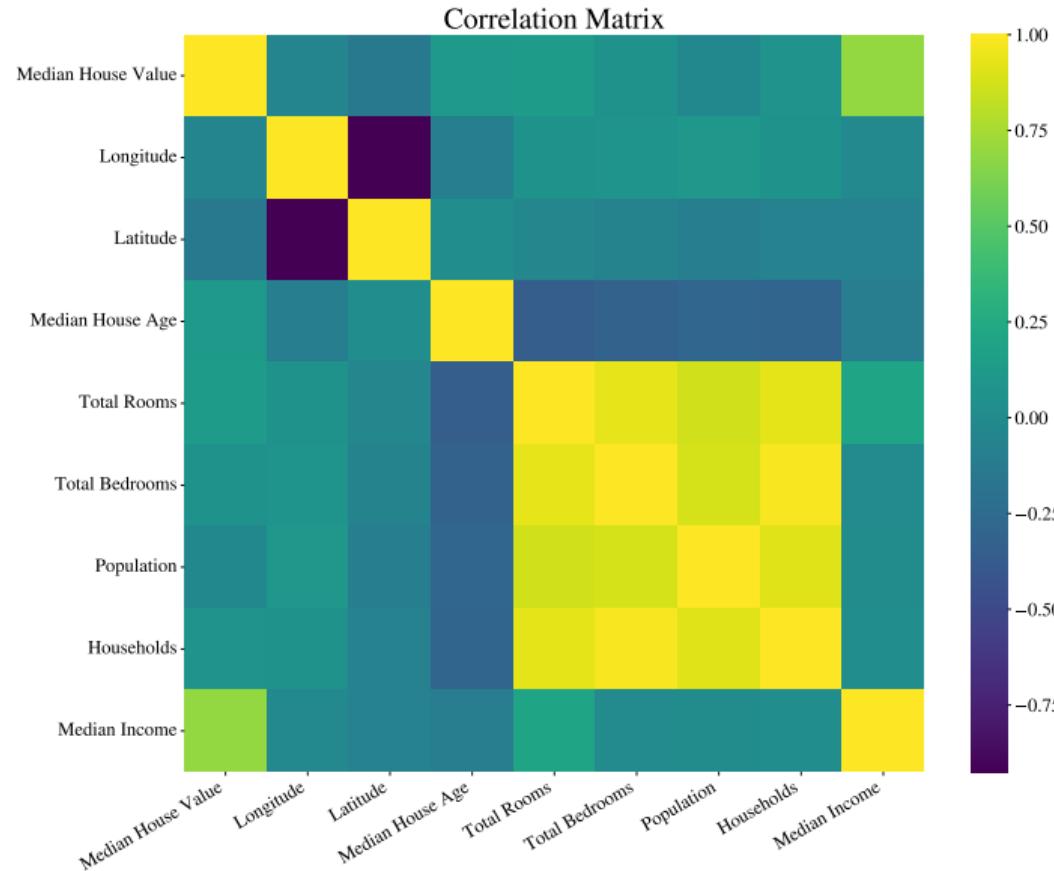
20428 rows × 10 columns

**9 features numeriche:** Prezzo; Coordinate; Età mediana delle case;  
Numero di: case, stanze, camere da letto; Popolazione; Reddito

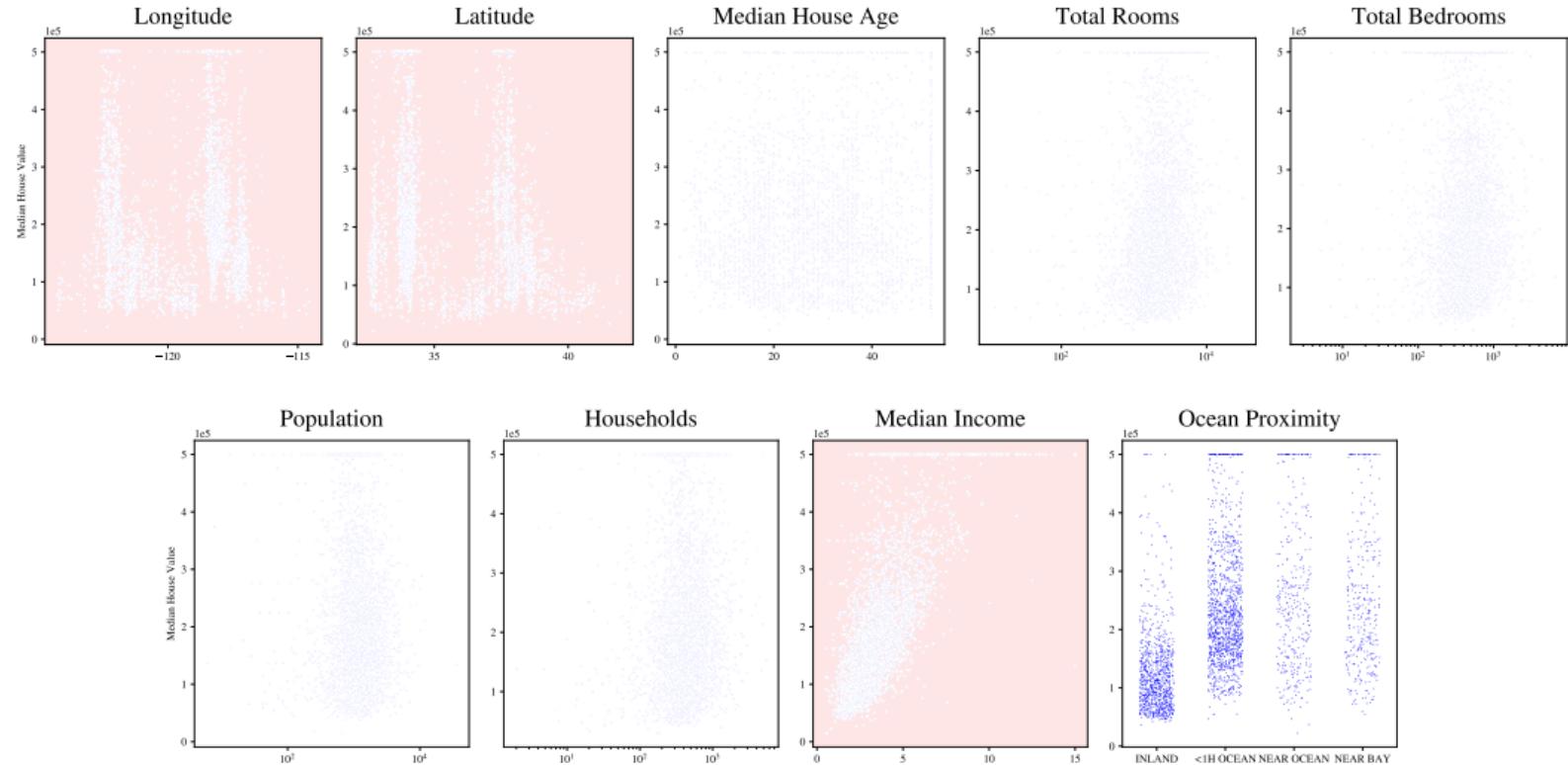
**1 feature categorica:** Prossimità dell'oceano

# Feature Selection

## Housing



# Housing Feature Selection



# Feature Selection

## Cars

	Selling Price	Year	Mileage [km]	Fuel	Seller Type	Transmission	Num of Previous Owners	Fuel Consumption [km/L]	Engine Size [cc]	Max Power [bhp]	Max Torque [Nm]	Seats
0	450000	2014	145500	Diesel	Individual	Manual	1	23.40	1248	74.00	190.00	5.0
1	370000	2014	120000	Diesel	Individual	Manual	2	21.14	1498	103.52	250.00	5.0
2	158000	2006	140000	Petrol	Individual	Manual	3	17.70	1497	78.00	124.46	5.0
3	225000	2010	127000	Diesel	Individual	Manual	1	23.00	1396	90.00	219.52	5.0
4	130000	2007	120000	Petrol	Individual	Manual	1	16.10	1298	88.20	112.70	5.0
...	...	...	...	...	...	...	...	...	...	...	...	...
8123	320000	2013	110000	Petrol	Individual	Manual	1	18.50	1197	82.85	113.70	5.0
8124	135000	2007	119000	Diesel	Individual	Manual	4	16.80	1493	110.00	235.20	5.0
8125	382000	2009	120000	Diesel	Individual	Manual	1	19.30	1248	73.90	190.00	5.0
8126	290000	2013	25000	Diesel	Individual	Manual	1	23.57	1396	70.00	140.00	5.0
8127	290000	2013	25000	Diesel	Individual	Manual	1	23.57	1396	70.00	140.00	5.0

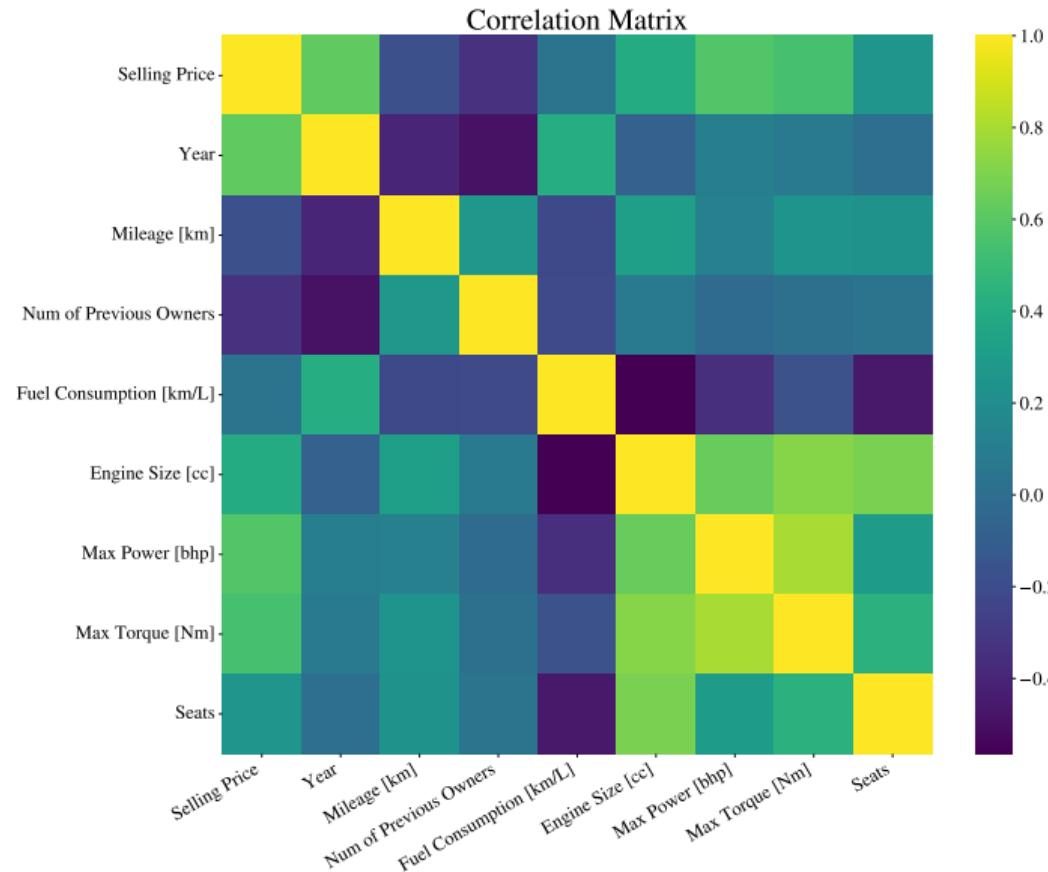
7270 rows × 12 columns

**9 features numeriche:** Prezzo; Anno; Chilometraggio; Numero di proprietari; Consumo; Cilindrata; Potenza e Coppia massime; Posti

**3 features categoriche:** Carburante; Venditore; Trasmissione

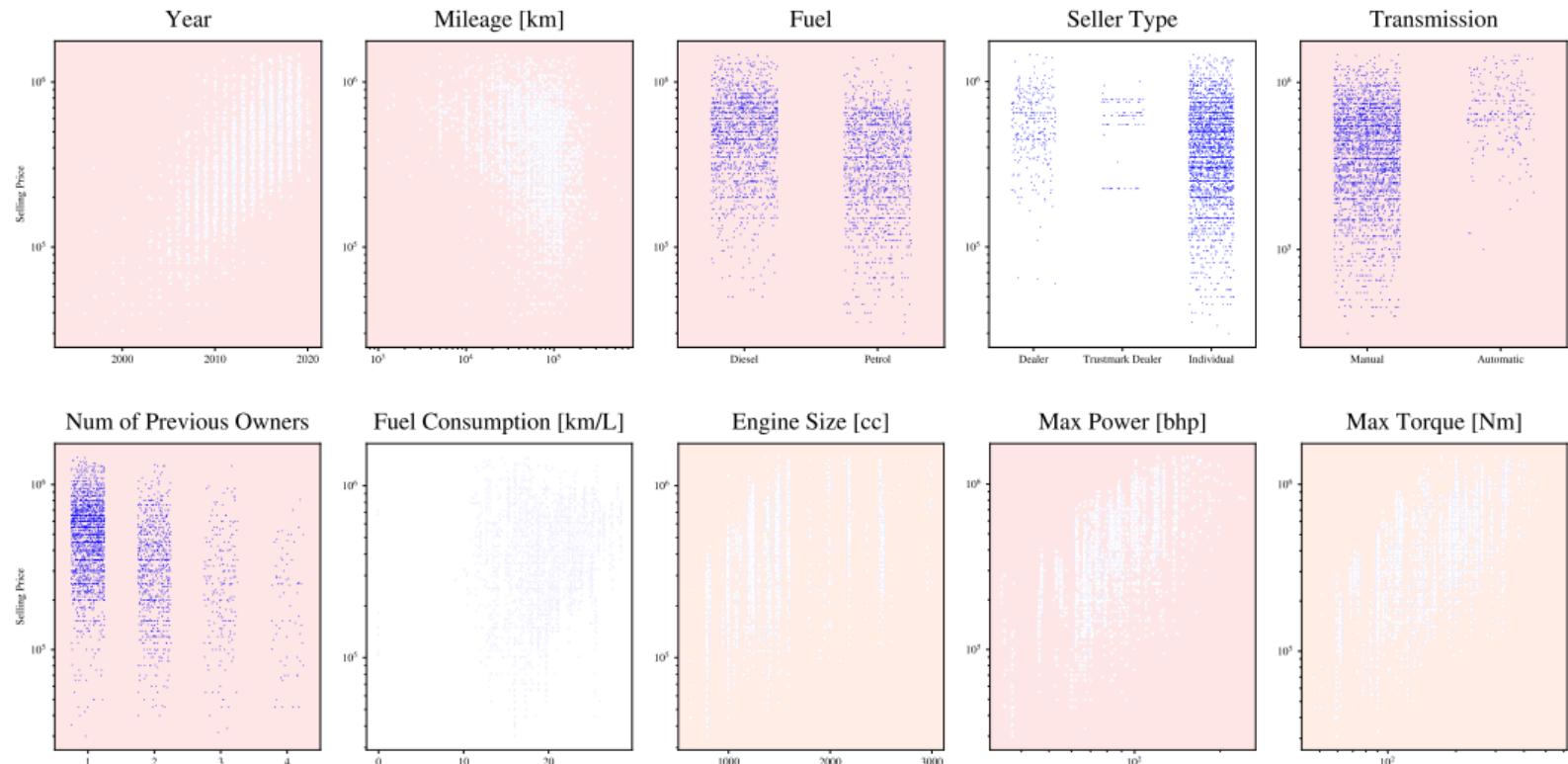
# Feature Selection

## Cars



# Cars

## Feature Selection

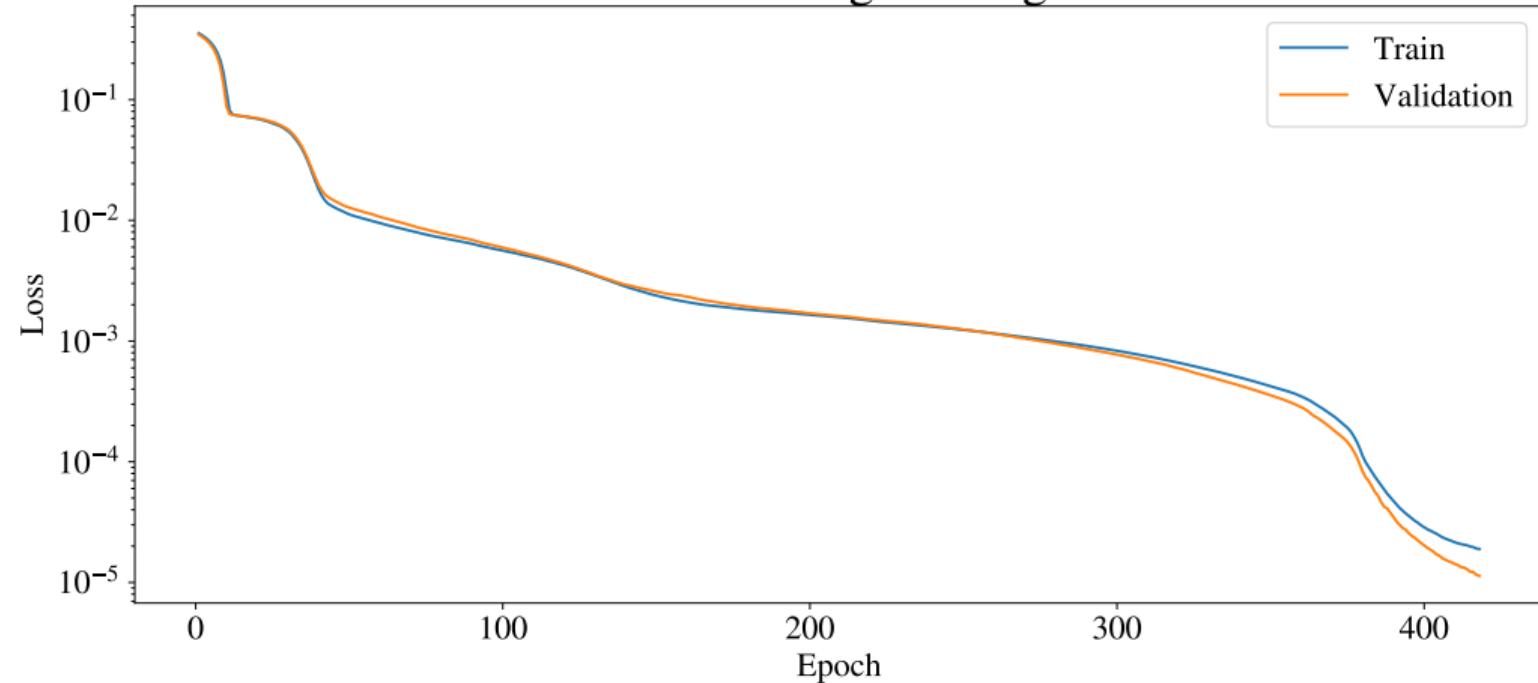


# Loss History

# Training History

Test dataset - FFNN

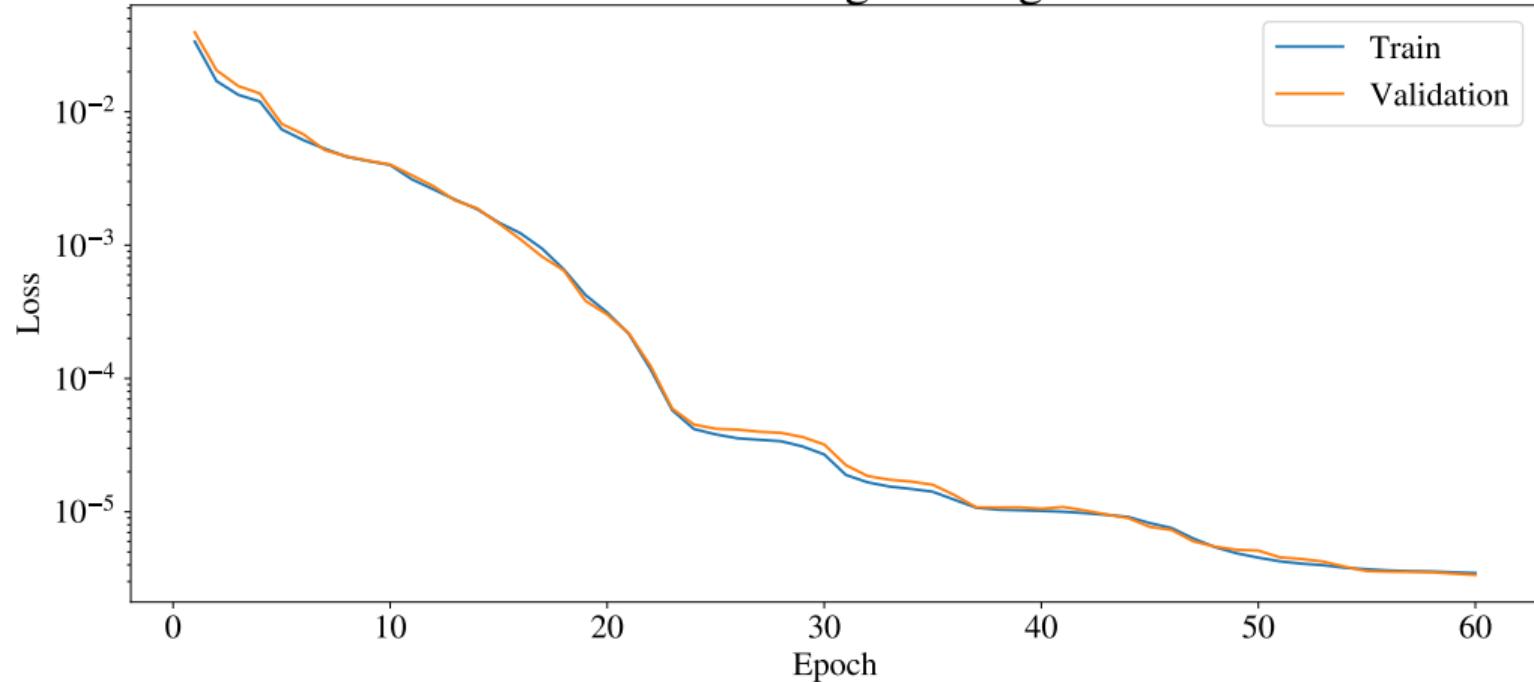
Loss during training



# Training History

## Test dataset - VQR

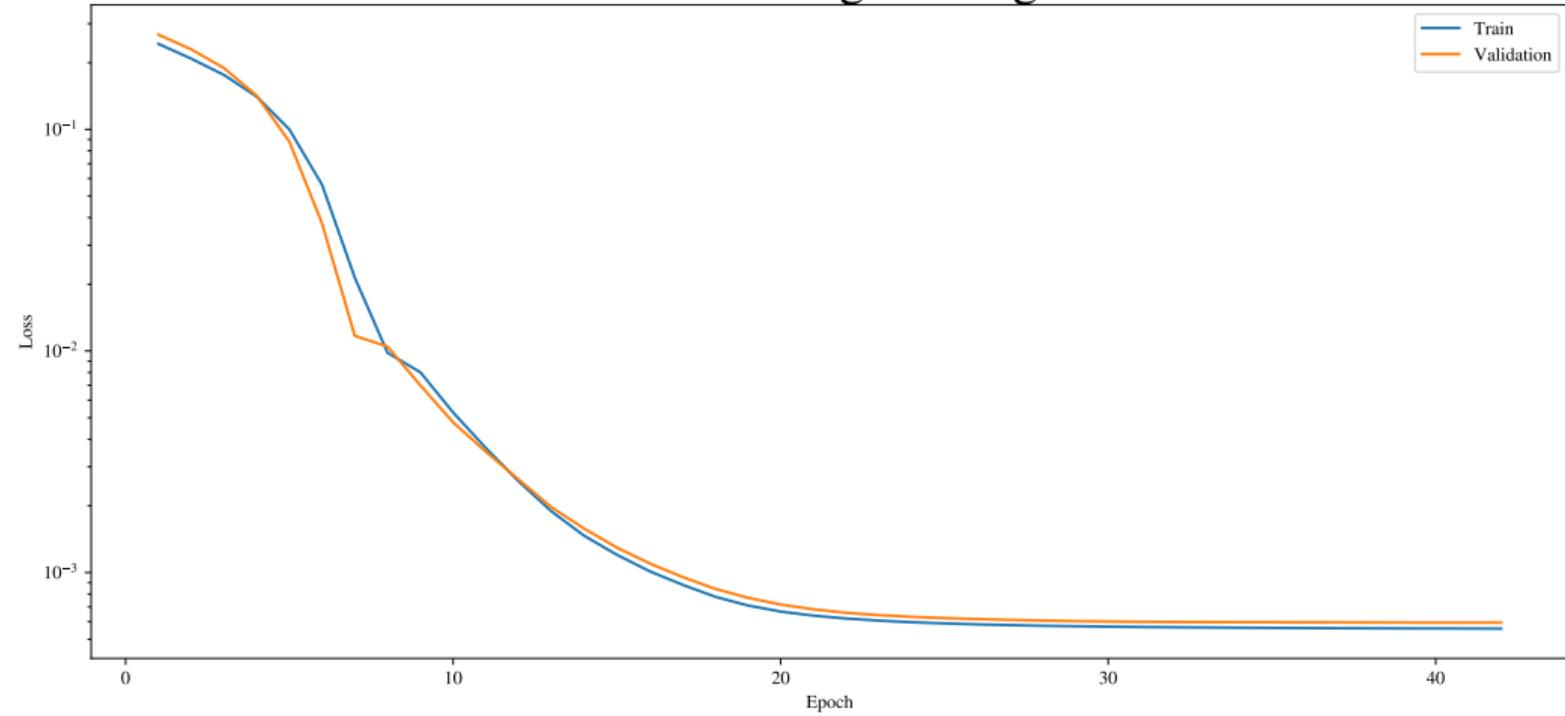
Loss during training



# Training History

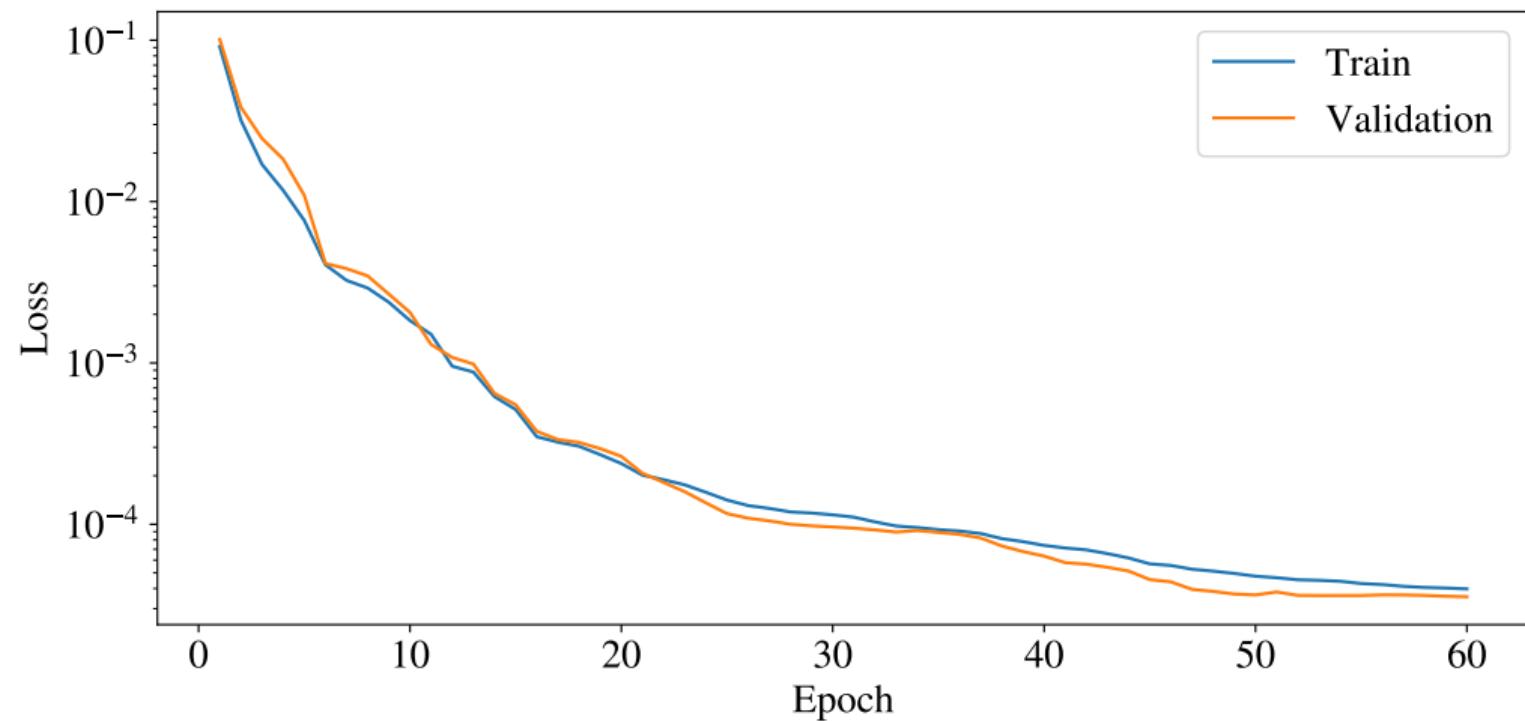
CO<sub>2</sub> 1var - FFNN

## Loss during training



# Training History

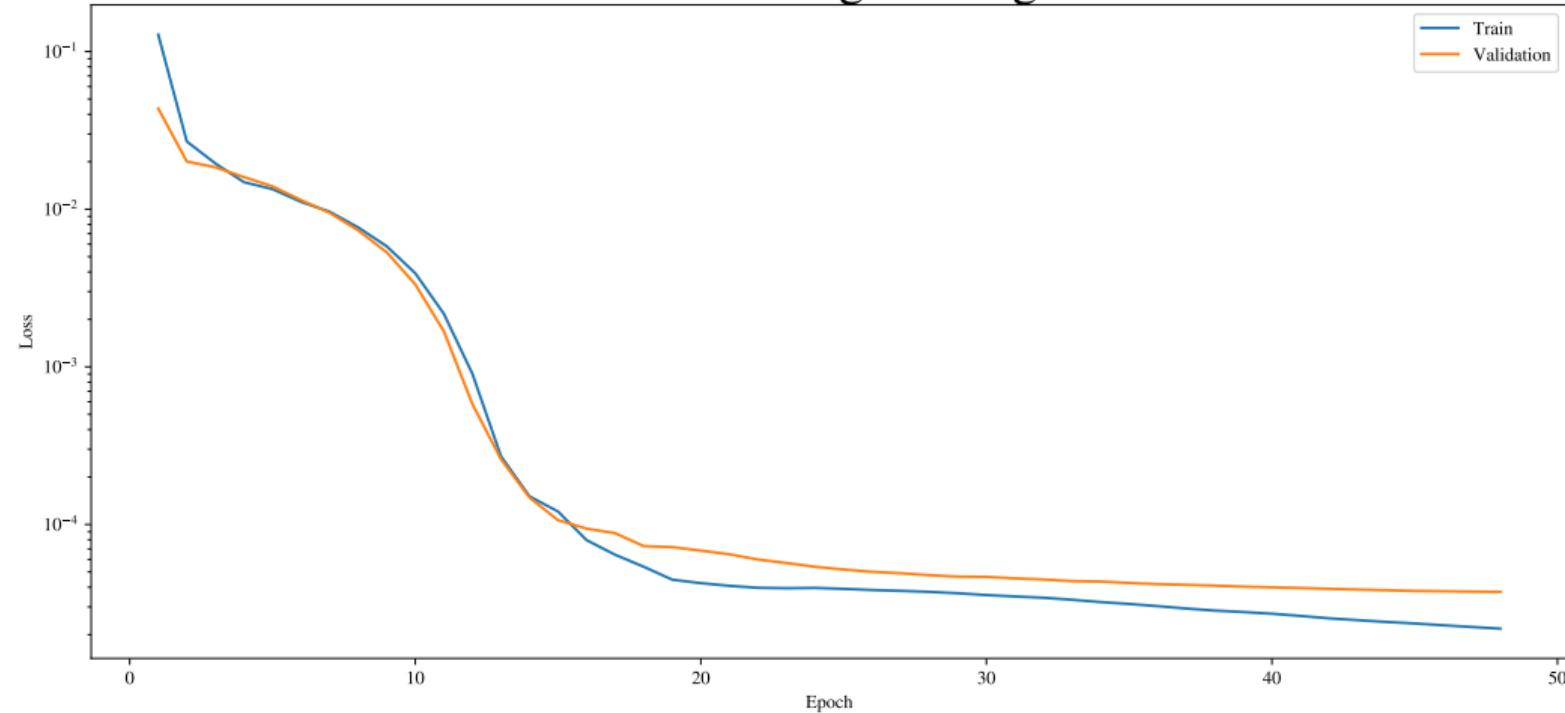
CO<sub>2</sub> 1var - VQR



# Training History

## CO<sub>2</sub> 2var - FFNN

### Loss during training



# Training History

CO<sub>2</sub> 2var - VQR

